The State of the Uniform: Attacks on Encrypted Databases Beyond the Uniform Query Distribution

Evgenios M. Kornaropoulos
UC Berkeley

Charalampos Papamanthou
University of Maryland

Roberto Tamassia
Brown University

***Abstract*—Recent foundational work on leakage-abuse attacks on encrypted databases has broadened our understanding of what an adversary can accomplish with a standard leakage profile. Nevertheless, all known value reconstruction attacks succeed under strong assumptions that may not hold in the real world. The most prevalent assumption is that queries are issued uniformly at random by the client. We present the first value reconstruction attacks that succeed *without any knowledge about the query or data distribution*. Our approach uses the search-pattern leakage, which exists in all known structured encryption schemes but has not been fully exploited so far. At the core of our method lies a support size estimator, a technique that utilizes the repetition of search tokens with the same response to estimate distances between encrypted values without any assumptions about the underlying distribution. We develop distribution-agnostic reconstruction attacks for both range queries and k-nearest-neighbor (k-NN) queries based on information extracted from the search-pattern leakage. Our new range attack follows a different algorithmic approach than state-of-the-art attacks, which are fine-tuned to succeed under the uniformly distributed queries. Instead, we reconstruct plaintext values under a variety of skewed query distributions and even outperform the accuracy of previous approaches under the uniform query distribution. Our new k-NN attack succeeds with far fewer samples than previous attacks and scales to much larger values of k. We demonstrate the effectiveness of our attacks by experimentally testing them on a wide range of query distributions and database densities, both unknown to the adversary.**

I. Introduction

In *searchable encryption* [15], [31], [41], a client encrypts a privacy-sensitive data collection and outsources an encrypted database to a server that can efficiently answer search queries without ever decrypting the database. Known constructions handle rich and expressive queries [17], [22] under the definitional framework of *structured encryption* (STE) [13]. For an overview of the area, see the survey by Fuller *et al.* [23].

To strike a balance between efficiency and privacy, structured encryption schemes reveal, by design, certain information about the query and its corresponding response—this is the so-called *leakage*. Despite cryptographic proofs guaranteeing that *nothing more is leaked* but what the designer allowed, the implications of the legitimately leaked information have not been fully grasped yet. The first generation of leakage-based attacks [8], [30], [45] focused on *query reconstruction* under various assumptions. The next generation of attacks [27], [32], [33], [34] supported *plaintext value reconstruction* by a server answering expressive queries, e.g. range and k-NN, on a one-dimensional database under strong assumptions about the query and/or data distribution. In this paper, we take the next step and demonstrate the first efficient reconstruction attacks for range and k-NN queries where the adversary has no knowledge about the query distribution or the underlying data.

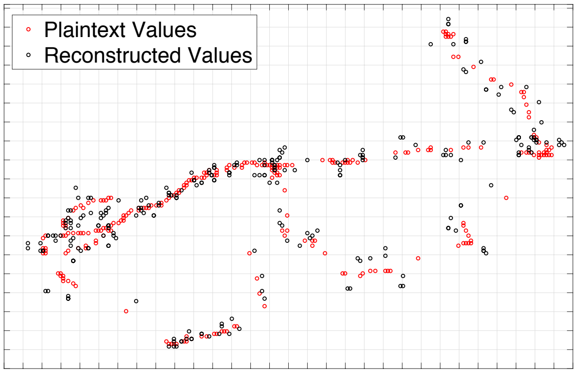


Fig. 1. Visual comparison between plaintext values of real-world private geolocation dataset Spitz (in red) and values reconstructed by our attack Agnostic-Reconstruction-KNN on k-NN queries under a *Gaussian distribution* and $k = 10$ (in black). Our attack achieves an approximate reconstruction (1) under a non-uniform query distribution and (2) with *half the queries* and larger k values compared to previous work [33].

A. Motivation and Approach

We overview the limitations of the four state-of-the-art attacks supported by a theoretical analysis and experimental evaluation [27], [32], [33], [34] and outline our new approach.

Uniform Query Distribution Assumption. The first value reconstruction attack for range queries was proposed by Kellaris-Kollios-Nissim-O'Neil (KKNO) [32]. It assumes that queries are issued *uniformly at random*. Lacharité-Minaud-Paterson (LMP) [34] studied the same problem for the special case of *dense* databases—this is a simpler problem since reconstructing order is equivalent to reconstructing values. The work by Grubbs-Lacharité-Minaud-Paterson (GLMP) [27] gives three reconstruction attacks for range queries under different assumptions: attacks GeneralizedKKNO and ApproxValue assume an underlying *uniform query distribution*, extend the underlying ideas of KKNO, and present a new analysis on the query complexity; attack AOR-to-ADR does not assume uniform queries but assumes that the attacker knows *both the query distribution and an approximation of the data distribution*. Kornaropoulos-Papamanthou-Tamassia (KPT) [33] propose reconstruction attacks for k-nearest neighbor queries under the *uniform query*

TABLE I
ASSUMPTIONS OF STATE-OF-THE-ART VALUE RECONSTRUCTION ATTACKS AND OUR NEW ATTACKS

Value Reconstruction Attack Algorithms	Query Type	Assumptions				Exploited Leakage	
		Query Distribution	Data Values in a Fixed Region	Known Data Distribution	Dense Database	Search-Pattern Leakage	Access-Pattern Leakage
KPT [33]	k-NN	Uniform	-	-	-	-	●
KKNO [32]	Range	Uniform	-	-	-	-	●
LMP [34]	Range	**Agnostic**	-	-	●	-	●
GLMP [27] GENERALIZEDKKNO	Range	Uniform	-	-	-	-	●
GLMP [27] APPROXVALUE	Range	Uniform	●	-	-	-	●
GLMP [27] AOR to ADR	Range	Known	-	●	-	-	●
This Work	k-NN & Range	**Agnostic**	-	-	-	●	●

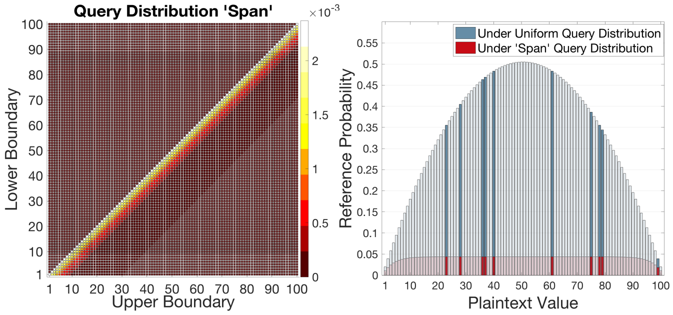
distribution. The above attacks, summarized in Table I, set the foundations for understanding the implications of leakage but only succeed under *strong assumptions* that potentially do not hold in the real world, e.g., uniform query distribution. Thus, the following question still remains open:

> *"Is it possible to devise attacks that reconstruct an approximation of the plaintext values without any knowledge about the query distribution or the data distribution?"*

Our work answers this question in the affirmative and presents reconstruction techniques that are *query and data distribution agnostic*. The key to achieve such a generalization lies in the *search-pattern leakage* which is revealed in all known STE schemes [23] but has been overlooked so far. See Figure 1 for an illustration of the quality of our reconstruction.

Fundamental Limitations of Current Range Attacks. A natural approach for answering the above question would be to extend existing algorithmic techniques to work for arbitrary query distributions. To explore this possibility, we first give a high-level intuition of the range reconstruction attacks KKNO, GENERALIZEDKKNO, and APPROXVALUE. Through the *access-pattern leakage*, which appears in the vast majority of STE schemes, the attacker can see *which* and *how many* queries return a given encrypted record. Assume the attacker knows the space of possible plaintext values, e.g., values from 0 to 100 representing attribute age. If range queries are *generated uniformly*, the attacker expects values in the middle (e.g. age $= 50$) to be returned more often than values towards the ends (e.g. age $= 1$). Formally, the *reference probability* of a value v captures the likelihood that value v will be returned in a response to a query. It is defined as $\sum_{r \in \mathcal{R}_v} \Pr[r]$, where $\mathcal{R}_v$ is the set of ranges containing v and $\Pr[r]$ is the probability of querying range r. Reference probabilities can be easily pre-computed by an attacker who knows the query distribution.

The reference probability of plaintext values for two query distributions is shown with histograms in Figure 2(b). Given enough queries, the attacker computes the frequency of each encrypted value and finds the closest match of each frequency to a pre-computed reference probability. Each matched reference probability corresponds to a plaintext value which is returned as the reconstructed value. This frequency-analysis works well for the uniform query case because the reference probabilities (blue histogram) vary significantly over the universe of plaintext values, therefore, one can accurately map the observed frequencies to reference probabilities. However, *there are fundamental limitations when trying to extend this approach.*



(a) (b)

Fig. 2. (a) Heatmap of the Span distribution of range queries on values from 0 to 100, where the probability of query $[a, b]$ is proportional to $(N - b + a)^{25}$. (b) Reference probabilities of plaintext values under the uniform query distribution (blue histogram) and Span query distribution (red histogram). Reconstructing the values of an encrypted database, shown with solid bars, from their empirical reference probabilities, is easy under the uniform query distribution but hard under the Span query distribution.

For instance consider the Span range query distribution, inspired by a realistic behavior from a client that issues "short ranges", depicted as a heatmap in Figure 2(a), where the lower boundary of the range is on the Y-axis, the upper boundary is on the X-axis and the color of each square denotes the probability of issuing this query. One can visually confirm that queries around the diagonal, i.e., queries with short span, have brighter color, hence are more likely to be issued. The reference probability for the Span query distribution is shown with the red histogram in Figure 2(b). Note that the reference probabilities of 60% of potential plaintext values differ by less than 10^{-8}, i.e., the middle part of the red histogram is almost flat, and as a result, the adversary can not make an accurate mapping from observed frequencies to reconstructed plaintext values. Thus, the Span query distribution causes all state-of-the-art attacks to fail.

More generally, one can define query distributions where the reference probabilities are *identical* so no matter how many queries are observed, the adversary cannot distinguish between potential plaintext reconstructions in the information-theoretic sense. Interestingly, the fact that frequency-based attacks fail in "smooth" distributions is used as a form of mitigation by Lacharité-Paterson [35], who introduce multiplicities in the records and spread the frequency of among the copies. From the above example we see that for range queries we need a *radically different reconstruction approach* to generalize.

How Many Queries Return a Response? Taking a step back to rethink reconstruction attacks, there is a piece of information that has not been fully exploited to overcome the uniform query assumption. This is the *number of queries that return a given response*, r, among the possible range queries. Let x_r be the number of lower query boundaries that can potentially return response r, and let y_r be the number of upper boundaries that can potentially return the same response, r. The total number of queries that return response r is essentially $N_r = x_r \cdot y_r$, but more importantly, both x_r and y_r are *distances between consecutive encrypted values*. Therefore if we know N_r for all r we could set up a system of equations containing x_r and y_r and retrieve the distances between all values in the database, effectively computing the values themselves.

However, the exact values of N_r are not available. Our main approach lies in *estimating N_r using search-pattern leakage* (which is part of all known constructions [23]) and then setting up *carefully-crafted optimization problems* to retrieve an estimation of the underlying distances/values.

Harnessing Search-Pattern Leakage. The *search-pattern leakage* reveals to the adversary if two encrypted queries, called *search tokens*, are generated from the same query. Interestingly none of the aforementioned state-of-the-art attacks [27], [32], [33], [34] utilize the search-pattern leakage, considering it harmless. We argue that this leakage can be instead exploited. Suppose that 10^3 observed search tokens (not-necessarily distinct) return response r. If these 10^3 tokens are the same, we can make a probabilistic argument that there aren't that many queries that return r. On the contrary, if all 10^3 are distinct, then there are clearly at least 10^3 queries, and likely more, that return r. More formally, the problem of estimating the number of *unseen outcomes from the frequency of observed outcomes* is called *support size estimation* and it has a rich history [5], [24], [44]. We use non-parametric support size estimation techniques that make no assumptions about the underlying distribution to re-think reconstruction algorithms for encrypted databases. Our techniques *reconstruct very accurately for the challenging case of "smooth" query distributions* due to the fact that our attacks are based on the number of possible queries that return a response, a quantity that can be estimated even under flat frequencies, as we demonstrate in our experiments.

B. Our Contributions

The influential work by Kellaris *et al.* [32] posed as a challenging open problem the task of plaintext reconstruction for query distributions beyond the uniform. Another open problem from [32] is the task of plaintext reconstruction for short range queries since, as the authors highlight, these queries are "typically observed in practice". In this work, we resolve these open problems by utilizing *both* the search-pattern and the access-pattern leakage for range and k-NN queries on one-dimensional databases by introducing attacks that are agnostic to the query and the data distribution.

• **Handling Unknown Query Distributions.** We first describe how the adversary can achieve knowledge transfer from statistics and learning theory to reconstruct encrypted databases. By partitioning the multiset of observed token-response pairs (t, r), the adversary can study each partition separately and draw inferences about the number of possible tokens that return r. We benchmark the state-of-the-art non-parametric support size estimation techniques under various (unknown to the adversary) query distributions. Our experiments indicate that certain estimators are better under different query distributions so we propose a new modular approach to pick the best estimation for the sample in hand. We further derive analytical expressions for known high-order non-parametric estimators, which is of independent interest.

• **A New Approach for Range Queries.** Armed with techniques for estimating the number of queries that return a response, we develop a new machinery to approximately reconstruct an encrypted database. On a high-level, each estimation gives us information about two distances between encrypted values. But these estimations are made independently and with a different sample sizes. We propose an efficient new algorithm, AGNOSTIC-RECONSTRUCTION-RANGE, that is based on an unconstrained convex optimization problem so as to piece together the above independent estimations and output *estimated distances between consecutive values of the database*. Our modeling gives higher weight to estimations made after observing a larger number of queries. We test our attack under a variety of query distributions and database densities, and show it achieves reconstructions with good accuracy. Also, AGNOSTIC-RECONSTRUCTION-RANGE outperforms GENERALIZEDKKNO for the majority of tested setups under the uniform query distribution, which is noteworthy because our algorithm is unaware of how the queries are issued and GENERALIZEDKKNO is fine-tuned for the uniform case.

• **Revisiting k-NN Queries.** For the problem of reconstruction from k-NN queries, we plug our support size estimators into the KPT algorithm to derive an estimation of the *length of the Voronoi segments* without relying on the uniform query distribution. Even though in theory this direct application is valid, due to the fact that for skewed query distributions the estimations are less accurate than in the uniform case, our initial experiments demonstrated that more often than not the resulting collection of estimated lengths is *not a Voronoi diagram* and thus KPT returns no reconstruction. To remedy this problem, we propose a new and efficient approach via formulating a constrained convex optimization problem that discovers the *minimum distortion* of the estimated lengths so as to force the lengths to *become* a valid Voronoi diagram. The formulation of KPT appears as a set of constraints in this new algorithm. Due to the minimum distortion insight, our proposed AGNOSTIC-RECONSTRUCTION-KNN always outputs a reconstruction as opposed to the all-or-nothing approach of KPT. Furthermore, since we don't explicitly build the set of all possible solutions, our approach scales to larger k compared with KPT. An illustration of a reconstruction for a real-world dataset of privacy-sensitive geolocation is shown in Figure 1. This reconstruction is achieved with *half the queries* compared to KPT, under a *Gaussian query distribution*, and with one-dimensional relative error of 0.08%.

II. Background

A *database* is a collection DB of n records $(id_i, val(id_i))$, $i = 0, \dots, n-1$ where id_i is a unique identifier and $val(id_i)$ is a value from the universe $[\alpha, \beta]$. We assume discrete values so that α, β, and $val(id_i)$ are integers and denote with $N = \beta - \alpha + 1$ the size of the universe. 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. We note though that our attacks can be extended to the case of non-injective mapping from records to values in which case the distance is 0 when consecutive records correspond to the same value. We call *density* of the database the percentage of values from the universe that are assigned to records. E.g., the *density assumption* studied by Lacharité *et al.* [34] corresponds to density 100%. A range query consists of two values $a \leq b$ from the universe and its response is the set of identifiers of the database records with values within interval $[a, b]$. A k-NN query consists of a value from the universe and its response is a set of k unordered identifiers that are closest to the query point, where k is fixed and decided at setup-time. We use the term *query* to refer to the plaintext query parameter(s) and the term *search token* to refer to the encrypted query parameter(s) that the client sends to the server. We define *access-pattern leakage* as the set of encrypted records that are retrieved as part of the response to a token. We define *search-pattern leakage* as the ability of the server to observe whether two tokens were generated from the same plaintext query. Although there are *response-hiding* STE schemes that minimize the access-pattern leakage by imposing a storage overhead, the widely-used constructions actually reveal the access-pattern for the sake of efficiency. To the best of our knowledge, all structured encryption schemes leak the search-pattern [23].

Assumptions. Our techniques have *no knowledge* about the query distribution, data distribution, or access to any auxiliary information about them. Our assumptions are as follows:

- **Static Database.** No updates, i.e., addition, deletions, take place once the database is encrypted.
- **Fixed Query Distribution.** We assume that the adversary issues independent and identically distributed (i.i.d.) queries with respect to a fixed query distribution. We emphasize that our adversary does not know *any information about the family or the parameters of the query distribution*.
- **Correctness.** We consider schemes where the response to the issued query is correct. We do not consider schemes that return missing responses or false positive responses, e.g., Logarithmic-SRC [17] and "over-covers" from [22].
- **One-dimensional Data Values.** We do not address encrypted databases for high-dimensional data [14].
- **Known Setup.** We assume that the adversary knows the number of encrypted values n, the size of the universe of values N and the endpoints of the universe α, β.
- **Injective Mapping of Search Tokens.** We assume that distinct queries, can be either a pair of values like the range queries or a single value like the k-NN, map to distinct search tokens. The injective mapping is satisfied, to the best of our knowledge, by all known STE encryption schemes.

Order Reconstruction. There is a plethora of techniques [27], [33], [34] in the literature that reconstructs the order of the encrypted values using only the access-pattern leakage. For simplicity of the exposition, we assume that the adversary can successfully reconstruct the order by using the appropriate algorithms from the above works and we instead focus on the problem of reconstructing the *plaintext values*. Thus, we treat the ordering as an input to our new value reconstruction algorithms and our techniques are not affected by how this ordering was constructed.

III. How to Exploit Search-Pattern Leakage

In this section, we introduce our main tool to reconstruct the plaintext values of an encrypted database *without any knowledge about the data or query distribution*. Given a fixed query distribution, the repetition of search tokens, i.e., search-pattern leakage, reveals information about *the total number of search tokens* that return a specific encrypted response. This key observation relates our attack to the extensively studied problem of estimating the support size of a distribution.

We first show how to partition token-response pairs and interpret them as samples from the unknown query distribution. Next, we benchmark two widely-used non-parametric estimators under various query distributions. Finally, we propose a new modular estimator for our attack. Since we obtain a different estimator per encrypted response, the next section shows how to glue the acquired estimations together to reconstruct the encrypted database in its entirety.

A. Conditional Probability Distributions over the Leakage

In this subsection, we show how an adversary that is given a multiset of m token-response pairs $D = \{(t_1, r_1), \dots, (t_m, r_m)\}$, can partition the tokens and analyze each group as a sample from a *conditional probability distribution*. By conditioning on the information observed from the access-pattern leakage, we group the information observed by the search-pattern leakage.

Remark 1. *Let $D = \{(t_1, r_1), \dots, (t_m, r_m)\}$ be the multiset of tokens and their corresponding response under an arbitrary token distribution. The mutliset of tokens with the same associated response, i.e., $D_i \coloneqq \{t_j | (t_j, r_i) \in D, r_i \subseteq \{id_0, \dots, id_{n-1}\}\}$, is a* sample from the conditional probability distribution $p_{T|R}(T = t | R = r_i)$.

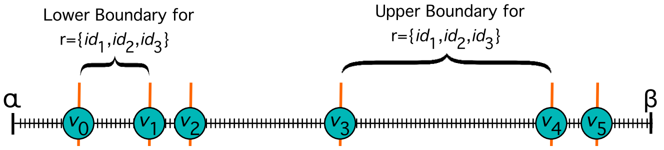


Fig. 3. To observe response $r = \{id_1, id_2, id_3\}$ the start of the query range must be in-between v_0 and v_1 and the end must be in-between v_3 and v_4. Thus, the total number of queries that return r is $(v_1 - v_0) \cdot (v_4 - v_3)$.

Range Queries. We recall again our assumption that the mapping from range queries to tokens is injective. However, we note that our attack can be applied also to structured encryption schemes that generate *multiple tokens per query* with no false positives. In this scenario the attacker creates a canonical ordering of the collection of tokens, e.g., by lexicographical-ordering, and treats their concatenation as a single token. Schemes with this property include the BRC and URC token generation presented in [17], as well as the cover selection approach presented in [22]. The partition of the token-response pairs is performed with respect to a specific response. Consider a database with values $\{v_0, \cdots, v_{n-1}\}$ from a universe $[\alpha, \beta]$. Since we do not consider schemes with false positives, the number of *distinct* tokens that return a given response $r = \{id_i, \cdots, id_j\}$ is equal the product $(v_i - v_{i-1}) \cdot (v_{j+1} - v_j)$, where v_{-1} and v_n refer to α and β, respectively. An example is depicted in Figure 3.

Remark 2. *For the case of range queries on an encrypted database the support size of the conditional distribution* $p_{T|R}(T = t|R = \{id_i, \ldots, id_j\})$*, where* $0 \leq i \leq j \leq n-1$*, is the product of (1) the distance between values* v_{i-1} *and* v_i *and (2) the distance between values* v_j *and* v_{j+1}*, i.e.,* $(v_i - v_{i-1}) \cdot (v_{j+1} - v_j)$*.*

k**-NN Queries.** A Voronoi diagram gives a natural partition of the query space for k-NN queries. Specifically each segment of the partition has the property that all the queries that land inside the segment have the same k nearest neighbors, i.e., the same response. It is known [33] that given a Voronoi diagram, the endpoints of each Voronoi segment correspond to bisectors between the values.

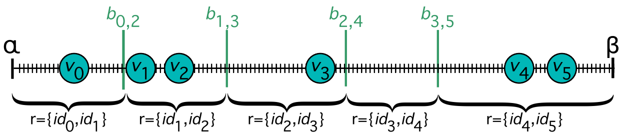


Fig. 4. Voronoi diagram of a database with 6 values $v_0, \ldots, v_5$ and 2-NN queries. Short vertical black lines indicate distinct queries and tall vertical green lines indicate bisectors $b_{i,i+2}$ for values v_i and v_{i+2}.

Figure 4 shows the Voronoi diagram for 2-NN queries on a database DB with values $v_0, \ldots, v_5$ from range $[\alpha, \beta]$. The bisectors of the diagram, $b_{i,i+2}$, partition the query points into intervals where queries yield the same response. E.g., all query points between bisectors $b_{1,3}$ and $b_{2,4}$ yield response $\{v_2, v_3\}$. In our scenario of an encrypted database, the response is a pair of identifiers. Accordingly, we define the following partition of query tokens for k-NN queries: a search token t belongs to group D_i if its response is $\{id_i, \ldots, id_{i+k-1}\}$, for $i \in [0, n-k]$. We recall here our assumption of an injective mapping from queries to tokens, i.e., we never map two distinct queries to the same token. Therefore, the probability distribution on k-NN queries transfers to the probability distribution on tokens.

Let T be a random variable whose possible values are the tokens for k-NN queries generated by the client under an arbitrary token distribution. Let R be a random variable whose possible values are the k-NN responses with respect to DB.

Remark 3. *For the case of* k*-NN queries on an encrypted database, the support size of the conditional distribution* $p_{T|R}(T = t|R = \{id_i, \ldots, id_{i+k-1}\})$ *is also the length of the corresponding Voronoi segment, i.e.,* $b_{i,i+k} - b_{i-1,i+k-1}$*.*

B. Estimate Support Size of Each Distribution

In this subsection, we show how to utilize the *frequency* of the observed search tokens so as to estimate the total number of search tokens that return a specific response r, i.e., estimate the support size of a conditional probability distribution with respect to r. In our approach, each response has a *different non-parametric estimator* that is "fine-tuned" for the specific conditional probability distribution. We focus on a *single response* but in the next section, we describe how an adversary can combine the estimations for different responses to achieve approximate reconstruction of the entire encrypted database. Furthermore, the estimation techniques described here are applied to both range and k-NN queries. To comply with the notation in the literature [44] on support size estimators, in this subsection N denotes the support size of a single query distribution, whereas in the rest of the paper N denotes the size of the universe of values, i.e., $N = \beta - \alpha + 1$.

Formulation. We assume a conditional probability distribution $p_{T|R}$ with respect to response r that contains N distinct search tokens observed with probabilities $\pi_i = (\pi_0, \ldots, \pi_{N-1})$. The adversary does not know the support size N or probabilities π_i. The main question we address is:

> *Given a sample* D *of* m *search tokens (with multiplicities) from* $p_{T|R}$*, what is the* ***total*** *number of search tokens in* $p_{T|R}$ *with non-zero probability?*

Let f_i be the number of search tokens that are observed i times in the sample. We briefly recall the terminology from [44]. The *fingerprint of sample* D is the vector $F = (f_1, f_2, \ldots, f_m)$, where $|D| = m$. Vector F is essentially the frequency of the frequencies. Then we can express the total number of *all distinct* search tokens as $N = f_0 + \sum_{i=1}^{m} f_i$ and the number of observed search tokens as $d = \sum_{i=1}^{m} f_i$. Similarly to [44], we call the *histogram of the query distribution* Q over the elements of $p_{T|R}$ the mapping $h_Q : (0, 1] \to [0, N]$, where $h_Q(\pi)$ is the number of $p_{T|R}$ elements that occur in probability mass function Q with probability π. Notice that the fingerprint is defined according to a sample while the histogram is defined according to the query distribution.

One Experiment Captures Multiple Distributions. We call a distribution property *symmetric*, or label-invariant, if it only depends on the histogram of the distribution. A symmetric property does not depend on which outcome maps to which probability. The next remark follows from Lemma 17 in [3].

Remark 4. *The* support size *of* $p_{T|R}$ *is a symmetric property.*

Jumping ahead, this important property comes into play in our evaluation. When we fix the query distribution in our experiments, we implicitly fix the conditional probability distributions too. The symmetric property implies that from the

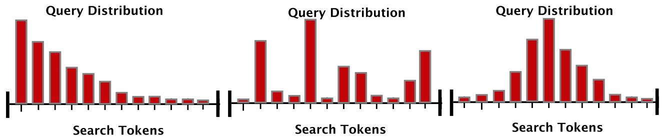


Fig. 5. An illustration of three query distributions with the same histogram. The result of a support size estimation is the *same* in all three cases.

point of view of the estimator, it *makes no difference which token maps to which fixed probability value*. Thus, the result of an experiment would be the same *for every assignment* of the chosen fixed probability values to tokens. As an example, the three probability mass functions presented in Figure 5 have *the same set of probabilities* but different labelings. Since the fingerprint is the same, the support size estimation on the ordered "towers" on the left gives the same estimation as the pmf in the middle or the bell-shaped pmf to the right.

Related Work. The problem of estimating the support size of a distribution has appeared in several fields in different forms. Examples include the estimations of the number of English words Shakespeare knew [21], the number of species in a population of plants or animals [7], and how many dies were used on an ancient coin [42]. As reviewing this large body of work is beyond the scope of this paper, we refer the reader to the following surveys [5], [12], [24]. We note that naive application of the estimators for the equiprobable case [29], [36] to settings with varying probabilities has been shown to give an estimation with negative bias [36].

In our work, instead of deploying parametric estimators that assume an underlying family of distributions, we use a more general *non-parametric* approach that is *distribution agnostic*.

The Jackknife Method. *Resampling techniques* are non-parametric methods of statistical inference that draw repeated subsamples from the original sample D. In this work we are interested in the *jackknife method* originally proposed by Quenouille in [40]. In certain scenarios it is not known how to compute an efficient unbiased estimator of a statistic of interest generally denoted as θ. Therefore given a biased estimator $\hat{\theta}$ for a statistic the jackknife approach *estimates the bias* via sampling with replacement from D. An estimate of the bias $\widehat{\text{bias}}_{Jack}$ can be used to correct the estimator as follows:

$$\hat{\theta}_{Jack} = \hat{\theta} - \widehat{\text{bias}}_{Jack}.$$

The resampling approach of the jackknife is the following: to form a new sample we leave one observation out so as to create the subsample $D_{(i)} = (d_1, \ldots, d_{i-1}, d_{i+1}, \ldots, d_m)$. We denote as $\hat{\theta}_{(i)}$ the estimation of θ that is computed based on $D_{(i)}$. The term $\hat{\theta}_{(.)}$ denotes the average of all possible leave-one-out estimations, i.e., $\hat{\theta}_{(.)} = \sum_{i=1}^{m} \hat{\theta}_{(i)}/m$. The jackknife bias is defined as:

$$\widehat{\text{bias}}_{Jack} = (m-1)(\hat{\theta}_{(.)} - \hat{\theta}) = (m-1)(\frac{1}{m}\sum_{i=1}^{m}\hat{\theta}_{(i)} - \hat{\theta}).$$

The multiplicative term $(m-1)$ in the above expression is rather counter-intuitive at first sight. One way to interpret this term is to assume that for a fixed m the expected value of the estimator $\hat{\theta}$ is the estimand plus a bias term of the form bias $= b_1(\theta)/m$. In this case we get:

$$E[\widehat{\text{bias}}_{Jack}] = (m-1)\left(E[\hat{\theta}] - \frac{1}{m}\sum_{i=1}^{m}E[\hat{\theta}_{(i)}]\right)$$
$$= (m-1)\left(\theta + \frac{b_1(\theta)}{m} - \theta - \frac{b_1(\theta)}{m-1}\right) = \frac{b_1(\theta)}{m} = \text{bias}$$

Therefore the *expectation of the bias estimate* is the true formula of the bias. The above exposition concerns the *first order jackknife estimator* since it corrects biases of the order $O(1/m)$. This approach can be generalized to formulate the *k-th order jackknife estimator* that results in a bias of the order $O(m^{-k-1})$. There is an inherit trade-off between the bias and variance, the higher the order of the jackknife estimator the smaller the bias and the larger the variance. Our estimators come directly from the work of Burnham and Overton [6], [7] and where originally proposed for estimating animal populations. The statistic that we are interested in is the total number of distinct classes N. The initial biased estimator $\widehat{N}$ is the number of distinct classes *observed in sample* D, i.e., $\widehat{N} = d = \sum_{i=1}^{m} f_i$. The following expressions present the "bias-corrected" formula of the originally biased estimator $\widehat{N}$. The order of the jackknife describes the level of bias correction applied. For a fixed sample size m the jackknife estimator of order i is a simple linear combination of the fingerprint $F = (f_1, \ldots, f_m)$. That is the i-th order jackknife estimator can be expressed as:

$$\widehat{N}_{J(i)} = \textstyle\sum_{k=1}^{m} \alpha_k^{(i)} f_k, \qquad (1)$$

where $\alpha_k^{(i)}$ coefficients are a function of the sample size m. The jackknife estimators for $\widehat{N}_{J(1)}, \widehat{N}_{J(2)}$, and $\widehat{N}_{J(3)}$ are:

$$\widehat{N}_{J(1)} = d + \tfrac{m-1}{m} f_1, \quad \widehat{N}_{J(2)} = d + \tfrac{2m-3}{m} f_1 - \tfrac{(m-2)^2}{m(m-1)} f_2,$$
$$\widehat{N}_{J(3)} = d + \tfrac{3m-6}{m} f_1 - \tfrac{(3m^2-15m+19)}{(m-1)m} f_2 + \tfrac{(m-3)^3}{(m-2)(m-1)m} f_3.$$

The derivation of the jackknife estimators $\widehat{N}_{J(i)}$ for $i \in [4, 10]$ appear in the Appendix, these analytical expressions may be of independent interest since they have not appeared before.

Selection of the Jackknife Order. Since we have we have the analytical expression of jackknife estimators $\widehat{N}_{J(i)}$, for $i \in [0, 10]$ an interesting question is how can we choose the appropriate order i given what we observed so far? To *tailor the order of the jackknife estimator* given the data in hand we deploy the order-selection technique originally proposed in [7] based on hypothesis testing. At a high-level this method tests the null hypothesis $H_i : E[\widehat{N}_{J(i+1)} - \widehat{N}_{J(i)}] = 0$ against $H_i' : E[\widehat{N}_{J(i+1)} - \widehat{N}_{J(i)}] \neq 0$ sequentially for $i \leq 10$ and choose the estimator $\widehat{N}_{J(i')}$ such that $H_{i'}$ is the first null hypothesis not rejected. We denote the above method for order selection as JACKKNIFE-SELFTUNE.

The Valiant-Valiant Estimator. The work by Valiant and Valiant [44] introduced a framework for rigorously estimating the histogram of a discrete probability distribution from a sample. Since we are using the estimator from [44] as is, we limit our exposition into a high-level description of the estimator and its guarantees and we refer the reader to the original manuscript [44] for the detailed description. The VALIANT-VALIANT estimator takes as an input a sample from an unknown distribution, creates the fingerprint and then

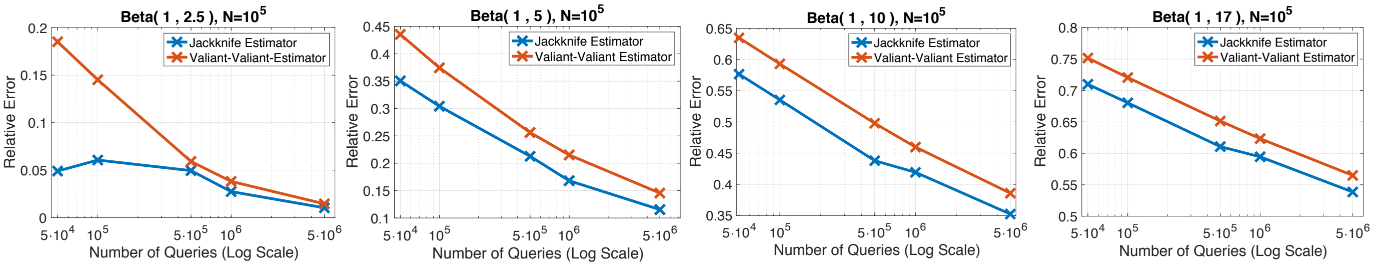


Fig. 6. Comparison of estimators JACKKNIFE-SELFTUNE and VALIANT-VALIANT with respect to their relative error in support size estimation.

computes a plausible histogram that might have produced the observed fingerprint. Because there are numerous histograms that explain equally well the observed fingerprint the authors propose a method that picks the "simplest" among them.

Theorem 1. *(Corollary 1.12 [44]) There exist absolute positive constants ζ, γ such that for any $0 < \epsilon < 1$, there exists N_ϵ such that for any $N > N_\epsilon$, given a sample of search tokens D of size $m > \frac{\gamma}{\epsilon^2}\frac{N}{\log N}$ sampled from any query distribution π over the domain of $p_{T|R}$ of size $|p_{T|R}| = N$, the* VALIANT-VALIANT *estimator outputs a $\hat{N}$ such that*

$$\Pr(|N - \hat{N}| \leq N\epsilon) \geq 1 - e^{-N^\zeta},$$

provided none of the probabilities in π lie in $(0, \frac{1}{N})$.

It is worth noting that the above guarantees are bounds on the convergence rate and not essential parameters for the VALIANT-VALIANT estimator. The algorithm itself *does not depend* on any of the above parameters and its only input is a sample D of any size. An alternative way to interpret the requirement that none of the probabilities in π lie in $(0, \frac{1}{N})$ is: the approximation guarantees only hold for all the search tokens with probabilities that are larger than $\frac{1}{N}$ and as a result there is no rigorous guarantee for detecting the tokens with probabilities within $(0, \frac{1}{N})$.

Evaluation of the Estimators. We conduct experiments to evaluate the performance of the estimators VALIANT-VALIANT and JACKKNIFE-SELFTUNE. The only input that the two non-parametric estimators take is a sample form an unknown query distribution and based on the frequency of the search tokens they estimate the support size. We compute the relative error of the support size estimation under different settings:

- *Query distribution.* We deploy a discretized Beta probability distribution $Beta(\alpha, \beta)$ defined under parameterizations that take values $\alpha = 1$ and $\beta = \{1, 2.5, 5, 10, 17\}$.
- *Scale of support size.* Chosen to be $N = 10^5$.
- *Number of observed search tokens.* Varying sample size.

We differentiate in our text between the α, β that denote the boundaries of the universe of values, see Section II, from the α, β used for the $Beta$ probability distribution by characterizing the latter as *parameters of the distribution*. Figure 7 shows the tested parameterizations of the Beta distribution. $Beta$ is defined under continuous interval $[0, 1]$ which we discretized into N segments of equal length. Parameter $\beta = 1$ gives the uniform distribution, parameter $\beta = 2.5$ gives an almost linear decay. For parameter $\beta = 10$, we have roughly a power law, i.e., the Pareto principle, where roughly 80% of the mass is distributed among 20% percent of the search tokens. This behavior has been recorded in a lot of real-world phenomena. To give some more concrete statistics, for parameters $\beta = 2.5, 5, 10, 17$ the percentages of search tokens that: (a) have probability less than $1/N$ are $54\%, 67\%, 77\%, 84\%$ and (b) have probability less than $1/N^2$ are $0.5\%, 12\%, 36\%, 54\%$, respectively. For each parametrization we tested $5{\cdot}10^3$ instances and in Figures 6 and 8 we report the average absolute relative error. We recall that even though our experiments are conducted over a fixed family of distributions, e.g., the beta distribution, by Remark 4 our observations apply to *any permutation* of the probability mass "towers" and thus cover a wide range of query distributions. Specifically a single benchmark covers all the $N!$ possible assignments of probabilities to labels/queries. As it can be seen in Figure 6 estimator JACKKNIFE-SELFTUNE is more accurate than VALIANT-VALIANT in the majority of the tested settings. The above measurements experimentally confirm the guarantees of Theorem 1 since a sublinear number of queries is enough to predict the existence of unobserved search tokens except the ones that have probability less than $1/N$. Another observation is that the maximum tested number of observed search tokens, i.e., $500N$, resulted in a relative error that is close to the percentage of search tokens with probability less than $1/N^2$.

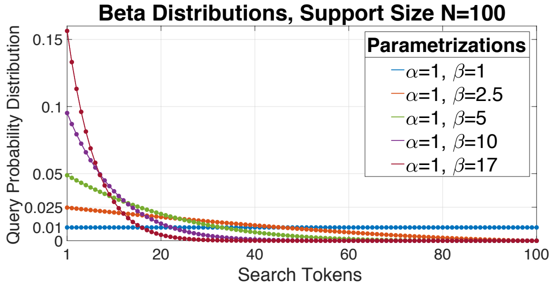


Fig. 7. Evaluation of the estimators is conducted under various query distributions parameterized as a Beta probability mass function.

Interestingly, for the case of uniform query distribution the VALIANT-VALIANT estimator is *significantly more accurate* when the number of samples is sublinear. Based on this observation we propose a "modular-estimator" to achieve the best of both worlds, an agnostic non-parametric estimator that deploys (1) the VALIANT-VALIANT when the query distribution

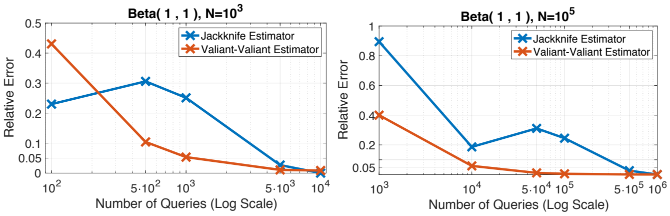


Fig. 8. Comparison of estimators under uniform query distribution.

is uniform and (2) the JACKKNIFE-SELFTUNE otherwise.

Modularity via Property Testing. Our estimator is Algorithm 1 (MODULAR-ESTIMATOR). The work of Goldreich and Ron [26] introduced a property testing [25] technique called *collision-probability tester* that given a sample from an unknown distribution it tests whether the sample originated from a distribution that is ϵ-afar from the uniform over $[1, N]$. Diakonikolas *et al.* [19] showed a tight upper bound on the sample complexity of $O(\sqrt{N}/\epsilon^2)$ which proves sample-optimality. The collision-probability tester takes as parameters the desired error ϵ the sample D and the support size N as an input. Unfortunately, in our setup we do not know N therefore in our algorithm we use the output of VALIANT-VALIANT as an approximation $\hat{N}$ to perform the collision-probability tester. Our approach is modular in the sense that different modules, i.e., estimators, are used for different "shapes" of query distributions. For concreteness we chose 0.1 as the threshold of the significance level of hypothesis testing, per recommendation of [7], and a fixed error ϵ for the collision-probability tester but these quantities can be tuned differently.

```
Algorithm 1: MODULAR-ESTIMATOR
Input: Multiset of $m$ search tokens $D$ sampled according to $p_{T|R}$
Output: Estimation of the support size $\hat{N}$
Deploy VALIANT-VALIANT estimator with input $D$ and get $\hat{N}_V$;
Compute number of collisions $c \leftarrow |\{j < k : k \in [2, m], t_j = t_k\}|$;
Set the error parameter for the tester $\epsilon \leftarrow 1/\hat{N}_V$;
if $c/\binom{m}{2} \leq (1 + 2\epsilon^2)/\hat{N}_V$ then  // collision prob. tester
    return $\hat{N}_V$ since it passed the tester
end
// Deploy JACKKNIFE-SELFTUNE;
Set number of unique tokens based on fingerprint $d \leftarrow \sum_{i=1}^{m} f_i$;
for $i \leftarrow 1$ to $9$ do
    Set $b_k \leftarrow \alpha_k^{(i+1)} - \alpha_k^{(i)}$, where $\alpha_k^{(i)}$ is the $k$-th coefficient of the jackknife estimator of order $i$, see Equation (1);
    $\hat{N}_{J(i+1)} - \hat{N}_{J(i)} \leftarrow \sum_{k=1}^{m} b_k f_k$          // Eq. (1);
    $\widehat{\text{var}}(\hat{N}_{J(i+1)} - \hat{N}_{J(i)}|d) \leftarrow \frac{d}{d+1}\left(\sum_{k=1}^{m}(b_k)^2 f_k - \frac{(\hat{N}_{J(i+1)} - \hat{N}_{J(i)})^2}{d}\right)$;
    Formulate the test statistic $T_i \leftarrow \frac{\hat{N}_{J(i+1)} - \hat{N}_{J(i)}}{\sqrt{\widehat{\text{var}}(\hat{N}_{J(i+1)} - \hat{N}_{J(i)}|d)}}$ for the null hypothesis $H_i : E[\widehat{N}_{J(i+1)} - \widehat{N}_{J(i)}] = 0$;
    Since $T_i$ follows approximately a standard distribution, we can derive its corresponding two-sided significance level, denoted as $P_i$;
    if $P_i > 0.1$ then
        return $\hat{N}_{J(i)}$ since the null hypothesis $H_i$ is not rejected
    end
end
```

IV. REVISITING DATA RECONSTRUCTION ATTACKS

In this section, we use the techniques from Section III to develop new reconstruction attacks on encrypted databases using *both* the search-pattern leakage and access-pattern leakage. Our reconstruction algorithm for range queries (Section IV-A) is significantly different from previous approaches. Our reconstruction algorithm from k-NN queries (Section IV-B) builds on previous work [33] but follows a different algorithmic strategy so as to (1) reduce the number of required samples and (2) scale for larger values of k. We experimentally demonstrate the accuracy of our reconstruction algorithms under various query distributions and densities of the database.

A. Reconstruction from Range Queries

Illustrative Example. We start by conveying the intuition of our range attack with an application on a simple database with only three values, $\{v_0 = 7, v_1 = 15, v_2 = 20\}$ from universe $[1, 30]$ shown in Figure 9. The distances between consecutive pairs, $L_i = v_i - v_{i-1}$, are $L_0 = 7$, $L_1 = 8$, $L_2 = 5$, $L_3 = 11$.

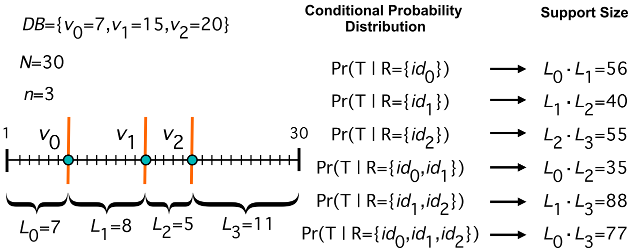


Fig. 9. Illustrative example of a database along with all the possible conditional probability distributions and their corresponding support size.

For simplicity, we consider first the restrictive scenario where the adversary has observed all possible range queries. In this case, there is *no need to estimate* the number of range-queries that return a specific response r', it is enough to count the number of unique queries that return r'. In other words, the adversary knows the exact support size for every conditional probability distribution $p_{T|R}(T|R = r')$. From Remark 2, the support size can be expressed as the product L_i, L_j for the appropriate pair i, j. The support sizes of all conditional distributions of this example are illustrated in Figure 9.

To compute the $n+1$ unknowns L_0, L_1, L_2, L_3, the adversary solves the following set of $\binom{n}{2}$ equations:

$$\begin{array}{lll} L_0 \cdot L_1 = 56 & L_1 \cdot L_2 = 40 & L_2 \cdot L_3 = 55 \\ L_0 \cdot L_2 = 35 & L_1 \cdot L_3 = 88 & L_0 \cdot L_3 = 77 \end{array} \quad (2)$$

One can apply the logarithmic function to transform the products to sums, i.e., $x_0 = \log(L_0), x_1 = \log(L_1), x_2 = \log(L_2), x_3 = \log(L_3)$. Then, using elementary row operations on the system of linear equations one can easily compute the echelon form and show that the rank of the matrix is $n+1$, thus there is a unique and exact reconstruction for the restrictive scenario where the adversary has seen all possible queries.

We now consider the more realistic, general scenario of an adversary who has observed a *subset* of all possible search tokens, as issued by the client under a fixed query distribution that is *unknown* to the adversary. From Observation 1, a token-response pair, (t', r'), can be seen as a *sample from the conditional probability distribution* $p_{T|R}(T|R = r')$. Thus, the first step of the attack is to partition the observed search tokens with respect to their returned responses, i.e., the conditional distribution they belong to, using the method of Section III.

The result of this partition gives a collection of multisets of search tokens. Each multiset is used to *estimate* the support size of the corresponding distribution. We denote with $\widehat{L}_{i,j}$ the *estimation* of the support size $L_i \cdot L_j$. We note here that some estimations should play a more central role in the overall reconstruction based on the fact that we have *observed more samples*. For example, the support size estimation of $p_{T|R}\,(T|R = r')$ from a sample of size 10 is less trustworthy than the support size estimation of $p_{T|R}\,(T|R = r'')$ from a sample of size 10^3. To capture this observation we model a minimization problem, where the "importance" of an estimate $\widehat{L}_{i,j}$ is expressed by a non-negative weight $w_{i,j}$.

```
Algorithm 2: AGNOSTIC-RECONSTRUCTION-RANGE
  Input: Multiset of range search tokens and their responses
         $D = \{(t_1, r_1), (t_2, r_2) \ldots, (t_m, r_m)\}$; ordering of the database
         records $I = (id_0, \ldots, id_{n-1})$; endpoints $\alpha$ and $\beta$ of the
         database universe; arbitrary positive constant $\epsilon$
  Output: Approximate reconstruction $\tilde{v}_0, \ldots, \tilde{v}_{n-1}$
  for every unique response $r$ in $D$ do
      Let $id_i \in r$ be the identifier of $r$ with minimum rank in $I$;
      Let $id_j \in r$ be the identifier of $r$ with maximum rank in $I$;
      Let $D_{i,j+1}$ be the mulitset of all the pairs in $D$ with response $r$;
      Let weight $w_{i,j+1} = \max\{\epsilon, |D_{i,j+1}|^2\}$;
      Run Algorithm 1 (MODULAR-ESTIMATOR) on the multiset of
        search tokens in $D_{i,j+1}$ to output estimated support size $\widehat{L}_{i,j+1}$;
  end
  Solve the system of linear equations below, obtained by setting the
    partial derivatives of Eq. (3) equal to zero:
    $$\begin{bmatrix} \sum_{j\neq 0} 2w_{0,j} & 2w_{0,1} & \cdots & 2w_{0,n} \\ 2w_{0,1} & \sum_{j\neq 1} 2w_{1,j} & \cdots & 2w_{1,n} \\ \cdots & \cdots & \cdots & \cdots \\ 2w_{0,n} & 2w_{1,n} & \cdots & \sum_{j\neq n} 2w_{j,n} \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ \cdots \\ x_n \end{bmatrix} = \begin{bmatrix} \sum_{j\neq 0} 2w_{0,j} \log(\widehat{L}_{0,j}) \\ \sum_{j\neq 1} 2w_{1,j} \log(\widehat{L}_{1,j}) \\ \cdots \\ \sum_{j\neq n} 2w_{n,j} \log(\widehat{L}_{n,j}) \end{bmatrix}$$
  Compute the approximated lengths as $L_0 = 2^{x_0}, \ldots, L_n = 2^{x_n}$;
  Scale $L_0, \ldots, L_n$ so as $\sum_{i=0}^{n} L_i = \beta - \alpha + 1$;
  Let $v_{-1} = \alpha - 1$;
  for $i = 0, \cdots, n-1$ do
      Let $\tilde{v}_i = \tilde{v}_{i-1} + L_i$;
  end
  return $\tilde{v}_0, \ldots, \tilde{v}_{n-1}$;
```

Reconstruction Algorithm. The goal of the proposed optimization is to assign values to the lengths $L_0, \ldots, L_n$ so as to minimize the weighted sum of squared errors. One option for the error function e is the difference between the two terms, i.e., $e_1(L_i, L_j) = (L_i \cdot L_j - \widehat{L}_{i,j})$. Another option for the error function is the logarithm of the ratio, i.e., $e_2(L_i, L_j) = \log\left((L_i \cdot L_j)/\widehat{L}_{i,j}\right) = \log(L_i) + \log(L_j) - \log(\widehat{L}_{i,j})$. If there is no sample to feed to the estimator to produce $\widehat{L}_{i,j}$, we assign default value $\widehat{L}_{i,j} = 1$, therefore the ratio in e_2 is well-defined since the denominator takes positive non-zero values. Notice that both e_1 and e_2 output 0 when the product of the unknowns is equal to the estimated quantity $\widehat{L}_{i,j}$. From experiments, we found that the error function of the $e_2(L_i, L_j)$ (log of ratio) has *superior reconstruction quality* in the majority of the cases compared to the error function $e_1(L_i, L_j)$. For simplicity, we define new unknowns $x_i = \log(L_i)$ for $i \in [0, n]$, which yields the following final unconstraint optimization problem:

$$\min_{x_0,\ldots,x_n} \sum_{i=0}^{n} \sum_{j=i+1}^{n} w_{i,j}(x_i + x_j - \log(\widehat{L}_{i,j}))^2 \tag{3}$$

We set weight $w_{i,j} = \max\{\epsilon, |D_{i,j}|\}$, where ϵ is an arbitrarily small positive value and $|D_{i,j}|$ is the number of tokens used for estimation $\widehat{L}_{i,j}$. The values $x_0, \ldots, x_n$ obtained from the solution of (3) are mapped to lengths as $L_i = 2^{x_i}$. As a final step, we scale the derived lengths $L_0, \ldots, L_n$ to sum to $N = \beta - \alpha + 1$ (total range of the database values).

Theorem 2. *The unconstrained quadratic optimization problem of Equation (3) with constant values $w_{i,j}, \widehat{L}_{i,j}$, and unknown values x_i, is a convex function and has a unique solution.*

The proof of Theorem 2 is in the Appendix. We derive the partial derivative with respect to x_i as:

$$\frac{\partial f}{\partial x_i} = \left(\sum_{j\neq i} 2w_{i,j}\right) x_i + \sum_{j\neq i} (2w_{i,j})\, x_j - \left(\sum_{j\neq i} 2w_{i,j} \log(\widehat{L}_{i,j})\right).$$

We find the global minimum by setting all partial derivatives equal to zero. Our reconstruction method from range queries, RANGE-RECONSTRUCTION, is shown in Algorithm 2.

Comparison with Attack GENERALIZEDKKNO [27]. We first compare the accuracy of the reconstruction of our attack, AGNOSTIC-RECONSTRUCTION-RANGE, to the accuracy of the state-of-the-art reconstruction attack GENERALIZEDKKNO, which is the most general (i.e., with fewest assumptions, e.g. only uniform queries) of the three attacks proposed by Grubbs *et al.* [27]. In this experiment, we generate $Q = 10^4$ range queries uniformly at random from the universe $[\alpha, \beta] = [1, 10^3]$. We randomly generate the values of the encrypted DB under various database densities. To assess the quality of the reconstruction, we use the *mean square error* (MSE) and the *mean of absolute error* (MAE) between the original and the reconstructed database. We note here that MSE gives a *higher penalty to reconstructed values with larger error*.

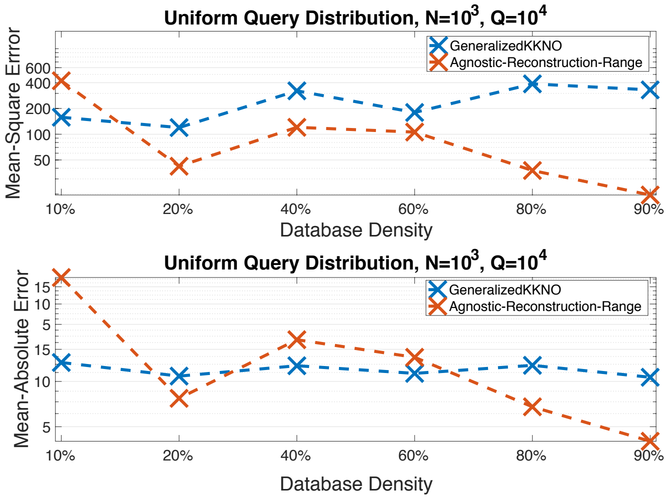


Fig. 11. Comparison between GENERALIZEDKKNO and our attack, AGNOSTIC-RECONSTRUCTION-RANGE, under the uniform query distribution.

Recall that our algorithm is (1) not tailored to work well on a specific query or data distribution and (2) distribution

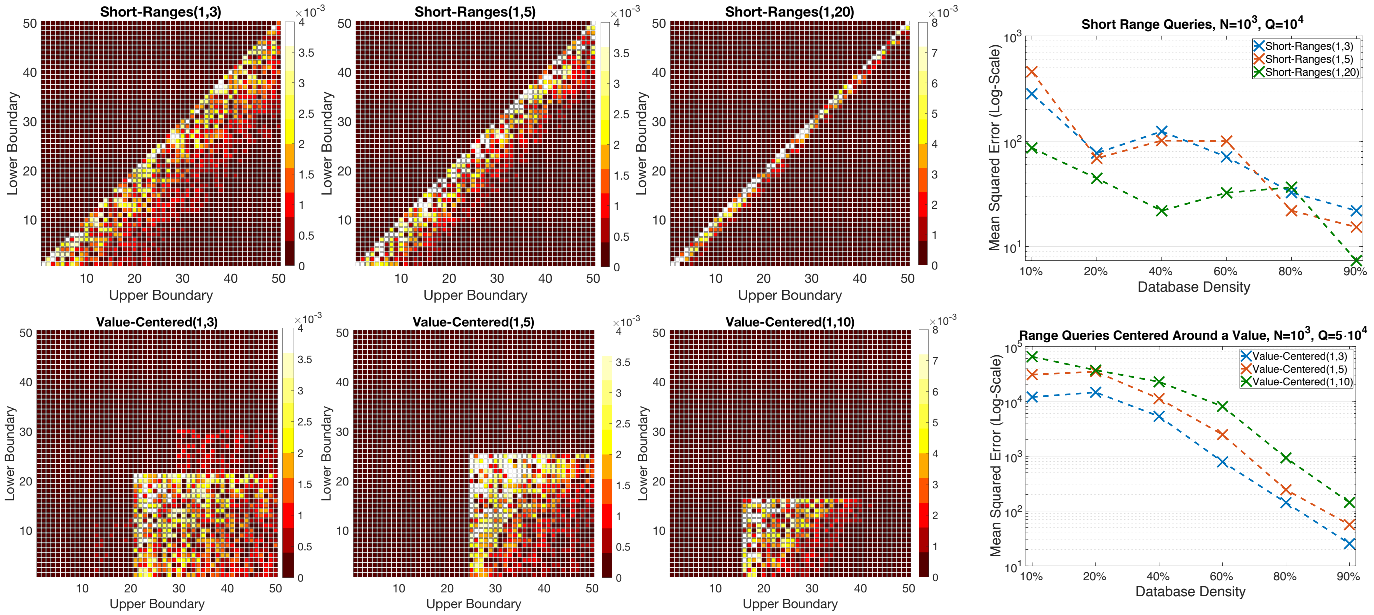


Fig. 10. Performance of our attack, AGNOSTIC-RECONSTRUCTION-RANGE, under parameterizations of query distributiona Short-Ranges and Value-Centered.

agnostic, i.e., does not need to know the data/query distribution. Hence, we would expect GENERALIZEDKKNO to have an inherent advantage in this experiment since it is specifically designed for uniform queries. The results of the experiment, shown in Figure 11, indicate that this is not the case: in terms of MSE, GENERALIZEDKKNO is $2.5\times$ to $17\times$ worse than our AGNOSTIC-RECONSTRUCTION-RANGE for densities from 20% to 90%, and in terms of MAE GENERALIZEDKKNO is comparable with our AGNOSTIC-RECONSTRUCTION-RANGE for densities from 15% to 75%.

We explain the experimental results as follows. The MAE quality metric is a first order statistic, therefore the large errors of GENERALIZEDKKNO are not penalized enough in the bottom plot of Figure 11. To explain why the performance of GENERALIZEDKKNO deteriorates, we note that this algorithm essentially maps the observed frequency to an expected frequency if the record were to have a fixed value. For dense databases, several records will appear together in many responses and as a result, will have very similar (if not identical) frequencies. This implies that multiple records map to the same plaintext value. The experiments confirm this behavior as GENERALIZEDKKNO tends to map multiple records to the same reconstructed value. To explain the outperformance of GENERALIZEDKKNO in the sparse regime, recall that the support size of each conditional distribution is *the product* of a pair of distances between database values. When the database is sparse, such distances are larger, hence the support size grows quadratically with the distance. Thus, the adversary needs to see more samples than the tested ones to increase accuracy.

Evaluation on Short Range Queries. In practical data analysis applications, focused short range queries are more likely to occur than exploratory long range queries. Also, a client who often issues long range queries would have limited benefits from outsourcing the database. Motivated by this observation, we have conducted experiments on short range queries. First, we explain how we generate short range query distributions and then we report on the experimental results.

Let $|R|$ be the number of all possible range responses. Specifically we generate a query distribution that we call Short-Ranges(α, β) as follows: Generate a Beta(α, β) distribution and discretize into $|R|$ equally spaced intervals. Recall that the cardinality of the universe of values is N. Then process the discretized values from left-to-right and add "noise" by multiplying each probability with a random number from $[0, 1]$ divided by $|R|$. After applying a normalization step, assign *in batches* the "noisy" Beta(α, β) probabilities to queries as follows: assigns the first N probabilities to queries whose range is a single value; assign the next $N - 1$ probabilities to queries whose range spans two values; continue up to the range query spanning the entire universe. This process gives higher probability to short range queries. The higher the value of parameter β, the larger the gap between the probabilities of short and long range queries. To understand how different Short-Ranges is from the uniform we note that for $N = 10^3$, the mean length of a sampled range query under the uniform is 333, which corresponds to 33% of the universe size. The query distributions Short-Ranges(1,3), Short-Ranges(1,5), and Short-Ranges(1,20) have mean length of 142, 90, and 23, which correspond to $14.2\%, 9\%$, and 2% of the universe, respectively.

In this evaluation, we chose parameter $\beta = \{3, 5, 20\}$ and $N = 10^3$. The upper row of Figure 10 shows the heatmap of the probability distribution for these three parameterizations but for a smaller universe. The Y-axis, resp. X-axis, corresponds to the lower boundary, resp. upper boundary, and the coloring of each square represents the probability of issuing this range query. As one can see the "bright" high-probability areas are around the diagonal. The MSE plot in Figure 10 shows the behavior of AGNOSTIC-RECONSTRUCTION-RANGE under different database densities. The distribution Short-Ranges(1,20) is a case where one would expect the reconstruction algorithm to be challenged due to the fact that only a few records are returned in each response. Interestingly, our reconstruction in Short-Ranges(1,20) is *significantly better* than the other distributions. To explain this, recall that the length of the range queries is really small which implies that the adversary only observes a small number of responses. So even though the majority of the total $\binom{N+1}{2}$ conditional probability distributions will not observe any query the small number of conditional distributions that are "active" will observe enough samples to get a very *accurate* estimation of their corresponding support size. The final step of the formulated convex optimization problem in Equation (3) combines the accurate estimations efficiently to derive the overall assignment of reconstructed values.

Evaluation on Queries Centered Around a Value. In this experiment we focus on range queries that are centered around a given value. Consider the real-world scenario of an encrypted database with medical data and assume that the client is a researcher who analyzes the medical profile of adolescents with asthma symptoms. We expect the majority of range queries issued by the client on the `age` attribute to have values within or near range $[13, 19]$ since this is the population of interest.

Inspired by the above scenario, we generate query distributions that we call Value-Centered(α, β), i.e., *tailored to contain a specific value* of the encrypted database. Similar to the generation process of the Short-Ranges query distributions, we discretize a beta pdf and multiply each probability of the pmf with a random number from $[0, 1]$ divided by $|R|$ and as a final step, we normalize. The difference from the previous experiment is how we assign the resulting probabilities to range queries. For Value-Centered we choose uniformly at random a value v'_1 of the underlying database. Processing again the probabilities-to-be-assigned from left to right, we assign the first batch of probabilities to the *range queries that return the chosen value* v'_1. As the next step, we sample without replacement another value v'_2 from the database and assign the next batch of probabilities to the ranges that return v'_2. This process continues until we have processed all n database values and we finally assign the remaining probabilities to the remaining range queries. The lower row of Figure 10 shows the heatmap of these distribution. The "bright rectangles" show that the range queries are "centered" around the value on the upper left corner of the rectangle. The ranges generated with Value-Centered(1,3) better explore the universe of values which allows our reconstruction attacks to achieve the smallest reconstruction error. The case of Value-Centered(1,10) assigns most of the high probabilities to a subset of the ranges that contain a single value therefore the majority of the universe is rarely explored. We report here that 14 out of the 120 runs of the Value-Centered(1,10) were unsuccessful because the queries did not explore the universe sufficiently. In general the query distribution Value-Centered(α, β) is makes the reconstruction more challenging than the previous distribution, a fact that is also supported by the observed MSE which is $100\times$ larger than the previous experiment. This reconstruction error can potentially be reduced by adding a small set of queries of exploratory nature scattered over the universe of values.

B. Reconstruction from k-NN Queries

In this subsection, we first discuss the limitations of the reconstruction attack AttackUnordered from k-NN queries by Kornaropoulos *et al.* [33]. Next, we introduce our method, which is scalable and supports reconstructions beyond uniform query distributions. Finally, we present experiments about the performance of our attack on synthetic and real-world datasets.

The two new ingredients of our reconstruction algorithm are: (1) use of support size estimators to compute an estimate of the length of each Voronoi segment *without any knowledge about the underlying query distribution*; and (2) formulation of an optimization problem that outputs a *minimal distortion* of the estimated lengths to transform them to a valid Voronoi diagram and thus become geometrically consistent. Previously proposed AttackUnordered [33] outputs no reconstruction in case the estimated lengths of the Voronoi segments are not geometrically consistent a case that we observed in most of our experiments when the query distribution is skewed.

Overview of AttackUnordered [33]. An insight from [33] is that even when the adversary observes all the possible queries, or else knowns the *exact* length of each Voronoi segment, it is impossible to achieve exact reconstruction of the encrypted database (see Theorem 2 in [33]). This is because there are arbitrarily many value assignments that have the same Voronoi diagram which implies that the reconstruction error comes from the combination of (1) the length estimation errors and (2) the choice of a reconstruction among the many valid ones. First, AttackUnordered estimates the length of a Voronoi segment $\textsc{Len}(\{id_i, \ldots, id_{i+k-1}\})$ as the frequency of a response $\{id_i, \ldots, id_{i+k-1}\}$ multiplied by the size of the universe of queries. This simple estimator is accurate under the assumption that the queries are generated uniformly at random. As shown in [33], any set of values that implies the observed Voronoi diagram satisfies three families of constrains:

- ordering constraints, i.e., $v_i < v_{i+1}$,
- bisector constrains, i.e., $(v_i + v_j)/2 = b_{i,j}$, and
- boundary constraints, i.e., $\alpha < v_0$ and $v_{n-1} < \beta$.

All the above constrains form a feasible region $\mathcal{F}$ of potential reconstructions, which is geometrically expressed as a k-dimensional convex polytope.

Limitations of AttackUnordered [33]. We identify here some limitations of the approach in [33] and later show how to overcome them. The length estimation in AttackUnordered can be performed *solely with the access-pattern leakage*, hence even though the adversary observes a wealth of information from the search-pattern, this information is not utilized. Also, algorithm AttackUnordered provides rigorous guarantees about the quality of the reconstruction, but this precision comes with a price. The experiments of [33] show that for a successful reconstruction, it is preferable to have (1) a large number of queries and (2) a small number of neighbors returned, k. Finally, the number of queries must be large enough so as the *estimated lengths are so accurate* *that they define a Voronoi diagram without any modification*. As an example, to achieve a reconstruction on the real-world Spitz dataset, the experiments in [33] observed more than 250 million queries. The proposed approach in this paper achieves a reconstruction on the same dataset with 2.5 million queries, a 100× smaller sample size. Overall, the exact approach of AttackUnordered [33] either succeeds with great accuracy or fails and outputs nothing. Additionally the technique in [33] requires the *explicit computation of the feasible region* by computing the vertices of the feasible region $\mathcal{F}$ which requires time that is linear to the number of vertices of $\mathcal{F}$. We note that a k-dimensional convex polytope has $O(2^k)$ vertices, therefore such an approach does not scale well to larger k values. Our new approach overcomes both of the above limitations and utilizes the search-pattern leakage.

Our Reconstruction Algorithm. Algorithm 3 (Agnostic-Reconstruction-KNN) outlines our attack from k-NN queries. A key insight is the use of the search-pattern leakage to estimate the length of each Voronoi segment *without any knowledge about the query distribution*. We build on the attack in [33] and extend it into two new directions. Instead of expecting a number of queries large enough to accurately estimate a valid Voronoi diagram, we compute the *minimum distortion* for each estimated length so as the *new "augmented" set of lengths comprise a valid Voronoi diagram*. We achieved this by adding *distortion variables* to the *offset variables* of [33] and introducing a convex optimization problem where the feasible region formulation from [33] forms the set of inequality constraints and the objective function expresses the minimization of the distortion. Finally, in order to scale to larger values of k, we don't require the explicit construction of the feasible region and instead output an arbitrary reconstruction from the feasible region of the augmented Voronoi diagram.

Observation 1 from Section III shows that an adversary with a sample D of search tokens and their responses can partition D with respect to each of the $n-k+1$ possible responses and form a collection of *samples from the conditional probability distributions*. From Remark 3 we know that the support size of a conditional probability distribution is the length of the corresponding Voronoi segment. Our algorithm deploys the Modular-Estimator to acquire an *estimation of the length* of each Voronoi segment without any assumptions about the query distribution, see Lines 2-6 in Agnostic-Reconstruction-KNN. After this step the estimated lengths, i.e., column vector $\widehat{l} = (\widehat{L}_0, \ldots, \widehat{L}_{n-k})$ is treated as constant.

We define one *distortion variable* δ_i per estimated length $\widehat{L}_i$, for $i \in [0, n-k]$. We derive the value assignment of these variables by solving a quadratic minimization problem where the objective function is the sum of the squares of δ_i, i.e., $\min \sum_{i=0}^{n-k} \delta_i^2$. This design choice captures our goal to compute the smallest possible distortion. We follow the footsteps of [33] and express the space of valid reconstructions with respect to offsets ξ_i from bisectors. Overall, the optimization formulation has $n-k+1$ unknowns for the distortion variables $\vec{\delta} = (\delta_0, \ldots, \delta_{n-k})$ and k unknowns for the offset variables $\vec{\xi} = (\xi_0, \ldots, \xi_{k-1})$, so a total of $n+1$ unknowns. The above objective function can be written as $\vec{x}^T M \vec{x}$, where $\vec{x}$ is the column vector from the concatenation of $\vec{\delta}$ and $\vec{\xi}$, and M is an all-zero matrix except the first $n-k+1$ elements of the main diagonal which have value 1. Since the matrix M is positive semidefinite, the objective function is a convex function.

Additionally the assignment of $\vec{\delta}$ and $\vec{\xi}$ should be such that the collection of augmented lengths, i.e., $\widehat{L}_i + \delta_i$ for $i \in [0, n-k]$, forms a Voronoi diagram. To express this goal we form four type of linear constraints for the optimization problem. The first type of constraints is the ordering constraints. These constraints can be written as $A \cdot [\vec{\delta}, \vec{\xi}]^T \leq B \cdot \widehat{l}$, where A is $(n-1) \times (n+1)$ matrix of constants and B is $(n-1) \times (n-k+1)$ matrix of constants. These matrices can be derived from the analytical formulas of Lemma 1 in the Appendix. The second type of constraints is the boundary constraints which guarantee that $\alpha < v_0$ and $v_{n-1} < \beta$, see Lemma 2 in the Appendix for the analytical formula. The third type of constraints guarantees that the offsets are positive, i.e., $\xi \geq 0$. Finally the fourth type of constraints is an equality constraint that guarantees that the augmented lengths sum to N, i.e., $\sum_{i=0}^{n-k} (\widehat{L}_i + \delta_i) = N$.

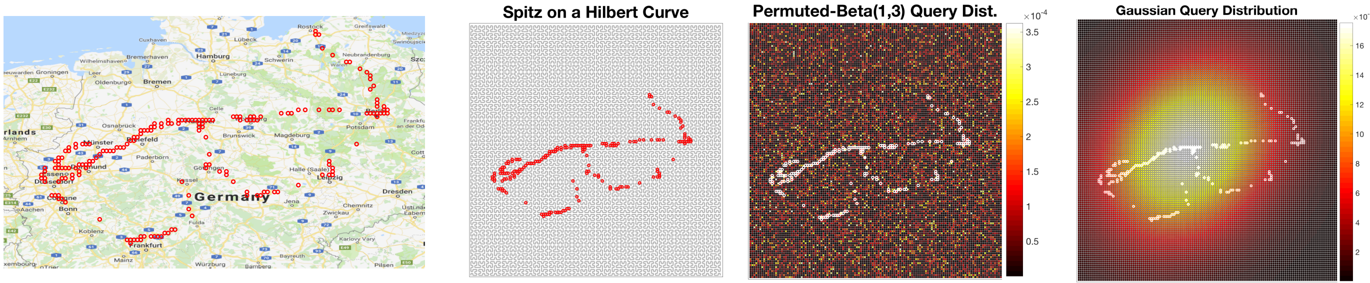
Algorithm 3: Agnostic-Reconstruction-KNN

```
Input: A multiset of $k$-NN search tokens and their responses
       $D = \{(t_1, r_1), (t_2, r_2) \ldots, (t_m, r_m)\}$, the ordering of the
       records $I = (id_0, \ldots, id_{n-1})$, as well as $\alpha, \beta, N$
Output: Approximate Reconstruction $\tilde{v}_0, \ldots, \tilde{v}_{n-1}$
1  Compute the left-to-right ordering $S$ of the responses, i.e., the Voronoi
   segments, using the ordering $I$ of the records.;
2  for every $r_i$ in $S$ from left-to-right do
3      Define $D_i$ as the mulitset with tokens from $D$ with response $r_i$;
4      Call Modular-Estimator with input the multiset of tokens $D_i$
       and store the estimated support size as $\widehat{L}_i$;
5  end
6  Define the vector of estimated lengths $\widehat{l} \leftarrow (\widehat{L}_0, \ldots, \widehat{L}_{n-k})$;
7  Solve the following convex optimization problem with unknowns the
   vector of distortions $\vec{\delta}$ and the vector of offsets $\vec{\xi}$:
```

$$\begin{aligned} \min_{\vec{\delta}, \vec{\xi}} \quad & \sum_{i=0}^{n-k} \delta_i^2 \\ \text{s.t.} \quad A \cdot \begin{bmatrix} \vec{\delta} \\ \vec{\xi} \end{bmatrix} & \leq B \cdot \widehat{l}, && \text{(ordering constraint)} \\ a_l^T \cdot \begin{bmatrix} \vec{\delta} \\ \vec{\xi} \end{bmatrix} & \leq b_l \cdot \widehat{l}, && \text{(lower boundary constraint)} \\ a_u^T \cdot \begin{bmatrix} \vec{\delta} \\ \vec{\xi} \end{bmatrix} & \leq b_u \cdot \widehat{l}, && \text{(upper boundary constraint)} \\ \vec{\xi} & \geq 0, && \text{(positive offsets constraint)} \\ \vec{\delta}^T \widehat{l} & = N, && \text{(sum of augmented lengths)} \end{aligned}$$

```
   where $A$ and $B$ are matrices of constant terms derived from the ordering
   constraints of Lemma 1, $a_l, b_l$ are vectors of constant terms for the lower
   boundary constraint derived from Lemma 2, and $a_u, b_u$ are vectors of
   constant terms for the upper boundary constraint derived from Lemma 2;
8  From the distortion vector $\vec{\delta}$ returned from the optimization problem and
   the estimated lengths $\widehat{l}$ we compute the augmented Voronoi diagram;
9  Given the above Voronoi diagram and the offset vector $\vec{\delta}$ returned from
   the optimization problem we derive the reconstructed database by
   substituting on the formulas of Lemma 5 in [33];
10 return $\tilde{v}_0, \ldots, \tilde{v}_{n-1}$;
```

Evaluation on the Spitz Dataset. In this experiment, we evaluate the performance of Agnostic-Reconstruction-KNN on a public real-world data set (also used in [33])



(a) (b) (c) (d)

Fig. 12. Real-world dataset Spitz of a privacy-sensitive geolocation trace: (a) data for Otober 1-31, 2009; (b) mapping of the points to a Hilbert curve which reduces the 2D data to 1D; (c) query distribution under attack, which consists of a permutation of a discretized Beta(α, β) distribution; (d) another query distribution under attack, which is a Gaussian centered at the city of Hannover, Germany.

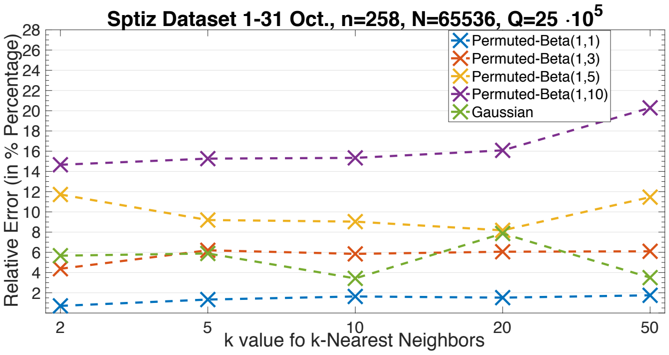


Fig. 13. Absolute relative error of Agnostic-Reconstruction-KNN for varying query distributions on the Spitz dataset.

containing the geolocation of politician Malte Spitz.[1] As in [33], we consider the geolocation data for the period October 1 to 31, 2009 and reduce the 2D data to 1D by deploying a Hilbert curve of order 8. The resulting discretized curve has universe of size $N = 65536$ and the dataset has size $n = 258$. The data is shown on a super-imposed map in Figure 12(a) and its mapping on a Hilbert curve is in Figure 12(b). The deployed query distribution is a discretized beta for varying parameters, similar to the experiments of the previous subsection but without any noise, i.e., *permuted over the universe of queries*. We illustrate this Permuted-Beta(α, β) query distribution with a heatmap on the superimposed data in Figure 12(c). Finally we test a Gaussian query distribution with mean centered at the city of Hannover in Germany and it is illustrated in Figure 12 (d).

The number of queries that the adversary observed is set to $Q = 25 \cdot 10^5$ which is 100× smaller sample size than the experiments conducted in [33]. Each attack was mounted 50 times and Figure 13 presents the average absolute relative error. Due to the new design of our reconstruction attack we were able to scale it to $k = 50$ an experiment that is not feasible from the approach followed in [33]. As it is expected the power-law like distribution Permuted-Beta(1,10) is the hardest to reconstruct due to the skewness and the sample size. Nevertheless the relative error ranges from 15% to 20% in this challenging scenario. The reconstruction under the Gaussian query distribution is accurate across all values of k.

Evaluation on Synthetic Dataset. We generated synthetic databases under varying densities and query distributions for $N = 10^3$ and $k = \{2, 5, 10, 20, 50\}$. Figure 14 shows the average of the mean absolute error of 100 repetitions with $Q = 10^5$. Note that for sparse databases, the distances between the values are larger, hence the offset variables have "more room" to deviate, which increases the size of the feasible region and as a result, the number of possible valid reconstructions. Another factor that increases the size of the feasible region is the increase of the value k, an intuition confirmed by the MAE for $k = 50$ even for the uniform case Permuted-Beta(1,1) which is easier to reconstruct. For densities larger than 20%, the reconstruction is usually within a distance of 20 from the plaintext value for all the tested query distributions.

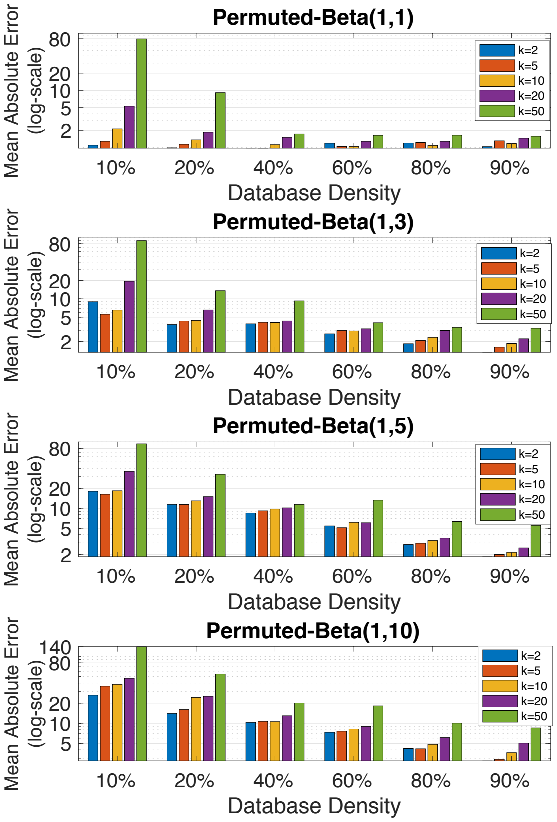


Fig. 14. Performance of Agnostic-Reconstruction-KNN for varying query distributions on synthetic data.

[1] www.zeit.de/datenschutz/malte-spitz-data-retention

V. Related Work

For encrypted single-keyword search [4], [9], [11], [15], [23], [31], [41], [43] the access pattern leakage of some leakage profiles is vulnerable to *query recovery attacks*, as opposed to the encrypted values. Specifically, Islam *et al.* [30], Cash *et al.* [8], and Zhang *et al.* [45] give *query-recovery* attacks under various assumptions. Encrypted systems with more expressive queries [39] rely on different cryptographic primitives, e.g., order-preserving encryption, and are vulnerable to data-recovery attacks [20], [28], [38] using only the setup leakage. In terms of efficiency there is a series of works [1], [2], [10], [16], [18] that study how the locality of searchable encryption affects the overall efficiency. We review in Section I state-of-the-art plaintext reconstruction attacks from from range queries [27], [32], [34] and from k-NN queries [33]. Recent work improves the asymptotic complexity of reconstruction from range queries under uniform query distribution using search pattern leakage [37].

VI. Discussion & Open Problems

The proposed attacks of this work are applied successfully to a wide range of realistic query distributions but it is worth noting that there exist distributions where our attacks fail to reconstruct. For example, the leakage observed in case all the probability mass is assigned to a single query is not enough to reconstruct the entire database. A similarly problematic case appears when the client issues queries that touch plaintext values from the first half of the universe, i.e. $[0, N/2]$, then the adversary would never see any leakage associated with the other half, i.e. $[N/2, N]$, and therefore fail to reconstruct the entire database. We leave as an open problem the task of characterizing the family of query distributions that are vulnerable to our proposed attacks. Another open problem is to analyze whether similar attacks can be mounted to the leakage derived from querying high-dimensional data. All the known attacks on ranges concern the quadratic scheme where the inverted index contains an entry for every possible range query. There exist constructions [17], [22] with much better storage efficiency than the quadratic scheme that also leak significantly less information. An open problem is to study whether it is possible to develop efficient reconstruction attacks for these advanced constructions for range queries. Moving on to schemes with strictly less leakage, the so-called response-hiding schemes store multiple copies of the same plaintext so as not to reveal that a record participates in multiple responses, i.e., they hide the overlap of records between responses via paying storage overhead. These schemes are immune to all the previous attacks as well as the attacks proposed in this work, an open problem is to analyze whether there exist any reconstruction attack for these schemes.

VII. Conclusion

This paper gives the first attacks on range queries and k-NN queries on encrypted databases that reconstruct the plaintext values *without any knowledge about the query or the data distribution*. Before our attacks, it was unclear whether such a general leakage-based reconstruction is possible as all previous approaches [27], [32], [33], [34] relied either on the uniform query distribution assumption or the assumption that the adversary knows both the query and the data distribution. These strong assumptions of previous attacks have given the false impression about their applicability and as a result, leakage-abuse attacks have been characterized in the past as “of theoretical interest”. The power and the generality of our reconstruction techniques, which overcome these strong assumptions, lie in the synergetic analysis of both the access-pattern and the search-pattern leakage. Experimental results demonstrate that an attacker can achieve accurate reconstruction under a wide variety of skewed query distributions and under various database densities without parametrizing the algorithms and without access to any auxiliary information.

VIII. Acknowledgments

This work was done while the first author was affiliated with Brown University and was supported in part by the U.S. National Science Foundation and by NIST.

References

[1] G. Asharov, M. Naor, G. Segev, and I. Shahaf, “Searchable Symmetric Encryption: Optimal Locality in Linear Space via Two-dimensional Balanced Allocations,” in *Proc. of the 48th ACM STOC*, 2016, pp. 1101–1114.

[2] G. Asharov, G. Segev, and I. Shahaf, “Tight Tradeoffs in Searchable Symmetric Encryption,” in *Proc. of the 38th CRYPTO*, 2018, pp. 407–436.

[3] T. Batu, L. Fortnow, R. Rubinfeld, W. D. Smith, and P. White, “Testing that Distributions are Close,” in *Proc. of the 41st IEEE FOCS*, 2000, pp. 259–269.

[4] R. Bost, “$\sum o\varphi o\varsigma$: Forward Secure Searchable encryption,” in *Proc. of the 23rd ACM CCS*, 2016, pp. 1143–1154.

[5] J. Bunge and M. Fitzpatrick, “Estimating the Number of Species: A Review,” *Journal of the American Statistical Association*, vol. 88, no. 421, pp. 364–373, 1993.

[6] K. P. Burnham and W. S. Overton, “Estimation of the Size of a Closed Population when Capture Probabilities vary Among Animals,” *Biometrika*, vol. 65, no. 3, pp. 625–633, 1978.

[7] ——, “Robust Estimation of Population Size When Capture Probabilities Vary Among Animals,” *Ecology*, vol. 60, no. 5, pp. 927–936, 1979.

[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] D. Cash and S. Tessaro, “The Locality of Searchable Symmetric Encryption,” in *Proc. of the 33rd EUROCRYPT*, 2014, pp. 351–368.

[11] J. G. Chamani, D. Papadopoulos, C. Papamanthou, and R. Jalili, “New Constructions for Forward and Backward Private Symmetric Searchable Encryption,” in *Proc. of the 25th ACM CCS*, 2018, pp. 1038–1055.

[12] A. Chao and C.-H. Chiu, *Species Richness: Estimation and Comparison*. American Cancer Society, 2016, pp. 1–26.

[13] M. Chase and S. Kamara, “Structured encryption and controlled disclosure,” in *Proc. of the 16th ASIACRYPT*, 2010.

[14] H. Chen, I. Chillotti, Y. Dong, O. Poburinnaya, I. Razenshteyn, and M. S. Riazi, “SANNS: Scaling Up Secure Approximate k-Nearest Neighbors Search,” Cryptology ePrint Archive, Report 2019/359, 2019, https://eprint.iacr.org/2019/359.

[15] 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.

[16] I. Demertzis, D. Papadopoulos, and C. Papamanthou, “Searchable encryption with optimal locality: Achieving sublogarithmic read efficiency,” in *Proc. of the 38th CRYPTO*, 2018, pp. 371–406.

[17] 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.

[18] I. Demertzis and C. Papamanthou, "Fast Searchable Encryption With Tunable Locality," in *Proc. of ACM SIGMOD*, 2017, pp. 1053–1067.

[19] I. Diakonikolas, T. Gouleakis, J. Peebles, and E. Price, "Collision-based Testers are Optimal for Uniformity and Closeness," *Electronic Colloquium on Computational Complexity (ECCC)*, vol. 23, p. 178, 2016.

[20] 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.

[21] B. Efron and R. Thisted, "Estimating the Number of Unseen Species: How Many Words Did Shakespeare Know?" *Biometrika*, vol. 63, no. 3, pp. 435–447, 1976.

[22] 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.

[23] 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.

[24] A. Gandolfi and C. C. A. Sastri, "Nonparametric estimations about species not observed in a random sample," *Milan Journal of Mathematics*, vol. 72, no. 1, pp. 81–105, Oct 2004.

[25] O. Goldreich, *Introduction to Property Testing*. Cambridge University Press, 2017.

[26] O. Goldreich and D. Ron, "On Testing Expansion in Bounded-Degree Graphs," *Electronic Colloquium on Computational Complexity (ECCC)*, vol. 7, no. 20, 2000.

[27] P. Grubbs, M. Lacharité, B. Minaud, and K. G. Paterson, "Learning to Reconstruct: Statistical Learning Theory and Encrypted Database Attacks," in *Proc. of the 40th IEEE S&P*, 2019, pp. 496–512.

[28] 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.

[29] B. Harris, "Statistical Inference in the Classical Occupancy Problem Unbiased Estimation of the Number of Classes," *Journal of the American Statistical Association*, vol. 63, no. 323, pp. 837–847, 1968.

[30] 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.

[31] S. Kamara, C. Papamanthou, and T. Roeder, "Dynamic Searchable Symmetric Encryption," in *Proc. of the 19th ACM CCS*, 2012, pp. 965–976.

[32] 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.

[33] E. M. Kornaropoulos, C. Papamanthou, and R. Tamassia, "Data Recovery on Encrypted Databases With k-Nearest Neighbor Query Leakage," in *Proc. of the 40th IEEE S&P*, 2019.

[34] 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.

[35] M. Lacharité and K. Paterson, "Frequency-Smoothing Encryption: Preventing Snapshot Attacks on Deterministically Encrypted Data," *IACR Transactions on Symmetric Cryptology*, vol. 2018, no. 1, pp. 277–313, Mar. 2018.

[36] R. C. Lewontin and T. Prout, "Estimation of the Number of Different Classes in a Population," *Biometrics*, vol. 12, no. 2, pp. 211–223, 1956.

[37] E. A. Markatou and R. Tamassia, "Full Database Reconstruction with Access and Search Pattern Leakage," in *Proc. of the 22nd ISC*, 2019.

[38] 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.

[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] M. H. Quenouille, "Approximate Tests of Correlation in Time-Series," *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 11, no. 1, pp. 68–84, 1949.

[41] D. X. Song, D. A. Wagner, and A. Perrig, "Practical Techniques for Searches on Encrypted Data," in *Proc. of the 21st IEEE S&P*, 2000, pp. 44–55.

[42] A. Stam, "Statistical Problems in Ancient Numismatics," *Statistica Neerlandica*, vol. 41, no. 3, pp. 151–174.

[43] E. Stefanov, C. Papamanthou, and E. Shi, "Practical Dynamic Searchable Encryption with Small Leakage," in *Proc. of the 21st NDSS*, 2014.

[44] G. Valiant and P. Valiant, "Estimating the Unseen: Improved Estimators for Entropy and Other Properties," *J. ACM*, vol. 64, no. 6, pp. 37:1–37:41, Oct. 2017.

[45] 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.

IX. Appendix

Proof of Theorem 2: We first show that the function in equation (3) is convex. Notice that the inner functions, i.e. $(x_i + x_j - \log(\widehat{L}_{i,j}))$, can be interpreted as convex functions, i.e. strict equality in the definition of convexity, and their composition with the quadratic function, i.e. $(x_i + x_j - \log(\widehat{L}_{i,j}))^2$, output a convex function. Furthermore it is known that the non-negative weighted sum of convex function preserves convexity which means that the function of (3) is convex. Due to convexity of (3) every local minima is global. The next step is to show that there exists a unique solution. Notice that the following matrix of coefficients M derived from the partial derivatives, also appears in Line 8 of Algorithm Agnostic-Reconstruction-Range, and is symmetric.

$$M = \begin{bmatrix} \sum_{j\neq 0} 2w_{0,j} & 2w_{0,1} & \dots & 2w_{0,n} \\ 2w_{0,1} & \sum_{j\neq 1} 2w_{1,j} & \dots & 2w_{1,n} \\ \dots & \dots & \dots & \dots \\ 2w_{0,n} & 2w_{1,n} & \dots & \sum_{j\neq n} 2w_{j,n} \end{bmatrix}$$

If we show that $\vec{y}^T M \vec{y} > 0$ for all vectors $\vec{y} \neq 0$, then the matrix is positive definite which implies that there is a unique solution. We want to show $\vec{y}^T M y > 0$. Thus:

$$\vec{y}^T M y > 0 \Rightarrow 2\vec{y}^T \tfrac{1}{2} M y > 0 \Rightarrow \vec{y}^T \tfrac{1}{2} M y > 0 \Rightarrow$$

$$\vec{y}^T \cdot \begin{bmatrix} \sum_{j\neq 0} w_{0,j} & w_{0,1} & \dots & w_{0,n} \\ w_{0,1} & \sum_{j\neq 1} w_{1,j} & \dots & w_{1,n} \\ \dots & \dots & \dots & \dots \\ w_{0,n} & w_{1,n} & \dots & \sum_{j\neq n} w_{j,n} \end{bmatrix} \cdot \vec{y} > 0 \Rightarrow$$

$$\left[y_0 \left(\sum_{j\neq 0} w_{0,j} \right) + \sum_{j\neq 0} y_j w_{0,j} \quad \dots \quad y_n \left(\sum_{j\neq n} w_{j,n} \right) + \sum_{j\neq n} y_j w_{j,n} \right] \cdot \vec{y} > 0 \Rightarrow$$

$$\sum_{i=0}^{n} \left(y_i^2 \left(\sum_{j\neq i} w_{i,j} \right) + y_i \sum_{j\neq i} y_j w_{i,j} \right) > 0 \Rightarrow$$

$$\sum_{0\leq i<j\leq n} \left(w_{i,j}(y_i^2 + y_j^2) \right) + \sum_{0\leq i<j\leq n} \left(2y_i y_j w_{i,j} \right) > 0 \Rightarrow$$

$$\sum_{0\leq i<j\leq n} w_{i,j} \left(y_i^2 + y_j^2 + 2y_i y_j \right) > 0 \Rightarrow$$

$$\sum_{0\leq i<j\leq n} w_{i,j} \left(y_i + y_j \right)^2 > 0$$

, which is true since w_i, j is always positive.

On the derivation of the Jackknife Estimators. Let m be the number of queries, k be the order of the jackknife estimator, d be the number of observed distinct queries, and $F = (f_1, \dots, f_m)$ be the fingerprint of sample D of queries. The formulation of the bias corrected jackknife estimator $\widehat{N}_{J(k)}$ of order k is given by:

$$\lambda_{m-j} = d - \binom{m}{j}^{-1} \sum_{t=1}^{j} \binom{m-t}{j-t} f_t,$$

$$\widehat{N}_{J(k)} = \frac{1}{k!} \left(m^k d + \sum_{j=1}^{k} (-1)^j \binom{k}{j} (m-j)^k \lambda_{m-j} \right).$$

Proof of Lemma 2. We replace every L_i with the term $L_i+\delta_i$ so as to consider the distortion variables $\delta_0,\dots,\delta_{n-k}$ we get: $\xi_0-\delta_0\le L_0$

Lower Boundary Constraint: Using Lemma 5 from [33]:

$$\alpha < v_0 \Rightarrow \alpha \le b_{0,k}-\xi_0 \Rightarrow \xi_0 \le b_{0,k}-\alpha$$
$$\Rightarrow \xi_0 \le \mathsf{Len}(\{id_0,\dots,id_{k-1}\}) \Rightarrow \xi_0 \le L_0$$

Upper Boundary Constraint:

- if $k \le n-1 < 2k$: Using Lemma 5 from [33] we get,
$$v_{n-1} = b_{(n-1)\bmod k,(n-1)\bmod k+k} + \xi_{(n-1)\bmod k} \Rightarrow$$
$$v_{n-1} = b_{(n-1)\bmod k,n-1} + \xi_{(n-1)\bmod k} \Rightarrow$$
$$v_{n-1} = \left(\alpha + \sum_{j=0}^{n-k-1}\mathsf{Len}(id_j,\dots,id_{j+k-1})\right) + \xi_{(n-1)\bmod k} \Rightarrow$$
$$v_{n-1} = \left(\alpha + \sum_{j=0}^{n-k-1} L_j\right) + \xi_{(n-1)\bmod k}$$
The upper boundary constraint is $v_{n-1} < \beta$.
$$v_{n-1} < \beta \Rightarrow$$
$$\left(\alpha + \sum_{j=0}^{n-k-1}(L_i+\delta_i)\right) + \xi_{(n-1)\bmod k} < \beta \Rightarrow$$
$$\xi_{(n-1)\bmod k} + \sum_{j=0}^{n-k-1}\delta_i < \beta - \alpha - \sum_{j=0}^{n-k-1} L_i \Rightarrow$$
$$\xi_{(n-1)\bmod k} + \sum_{j=0}^{n-k-1}\delta_i < \sum_{j=0}^{n-k} L_i - \sum_{j=0}^{n-k-1} L_i \Rightarrow$$
$$\xi_{(n-1)\bmod k} + \sum_{j=0}^{n-k-1}\delta_i < L_{n-k}$$
- if $2k \le n-1$: Using Lemma 5 from [33] we get,
$$v_{n-1} = (-1)^{\lfloor (n-1)/k-1\rfloor}(b_{(n-1)\bmod k,((n-1)\bmod k)+k} + \xi_{(n-1)\bmod k}) +$$
$$+\sum_{j=2}^{\lfloor (n-1)/k\rfloor}(-1)^{j+\lfloor (n-1)/k\rfloor} 2b_{((n-1)\bmod k)+(j-1)k,((n-1)\bmod k)+jk} \Rightarrow$$
$$v_{n-1} = (-1)^{\lfloor (n-1)/k-1\rfloor}\xi_{(n-1)\bmod k} +$$
$$+(-1)^{\lfloor (n-1)/k-1\rfloor} b_{(n-1)\bmod k,((n-1)\bmod k)+k} +$$
$$+\sum_{j=2}^{\lfloor (n-1)/k\rfloor}(-1)^{j+\lfloor (n-1)/k\rfloor} 2b_{((n-1)\bmod k)+(j-1)k,((n-1)\bmod k)+jk} \Rightarrow$$
$$v_{n-1} = (-1)^{\lfloor (n-1)/k-1\rfloor}\xi_{(n-1)\bmod k} +$$
$$+(-1)^{\lfloor (n-1)/k-1\rfloor}(\alpha + \sum_{j=0}^{(n-1)\bmod k}\mathsf{Len}(id_j,\dots,id_{j+k-1})) +$$
$$+\sum_{j=2}^{\lfloor (n-1)/k\rfloor} 2(-1)^{j+\lfloor (n-1)/k\rfloor}\cdot\left(\sum_{m=0}^{\substack{((n-1)\bmod k)\\+(j-1)k}}\mathsf{Len}(id_m,\dots,id_{m+k-1})\right) \Rightarrow$$
$$v_{n-1} = (-1)^{\lfloor (n-1)/k-1\rfloor}\xi_{(n-1)\bmod k} +$$
$$+(-1)^{\lfloor (n-1)/k-1\rfloor}(\alpha + \sum_{j=0}^{(n-1)\bmod k} L_j) +$$
$$+\sum_{j=2}^{\lfloor (n-1)/k\rfloor} 2(-1)^{j+\lfloor (n-1)/k\rfloor}\cdot\left(\sum_{m=0}^{\substack{((n-1)\bmod k)\\+(j-1)k}} L_m\right)$$
The upper boundary constraint is $v_{n-1} < \beta$.
$$v_{n-1} < \beta \Rightarrow$$
$$(-1)^{\lfloor (n-1)/k-1\rfloor}\xi_{(n-1)\bmod k} +$$
$$+(-1)^{\lfloor (n-1)/k-1\rfloor}(\alpha + \sum_{j=0}^{(n-1)\bmod k}(L_j+\delta_j) +$$
$$+\sum_{j=2}^{\lfloor (n-1)/k\rfloor} 2(-1)^{j+\lfloor (n-1)/k\rfloor}\cdot\left(\sum_{m=0}^{\substack{((n-1)\bmod k)\\+(j-1)k}}(L_m+\delta_m)\right) < \beta \Rightarrow$$
$$(-1)^{\lfloor (n-1)/k-1\rfloor}\xi_{(n-1)\bmod k} + (-1)^{\lfloor (n-1)/k-1\rfloor}\sum_{j=0}^{(n-1)\bmod k}\delta_j +$$
$$+\sum_{j=2}^{\lfloor (n-1)/k\rfloor} 2(-1)^{j+\lfloor (n-1)/k\rfloor}\cdot\left(\sum_{m=0}^{\substack{((n-1)\bmod k)\\+(j-1)k}}\delta_m\right)$$
$$< \beta - (-1)^{\lfloor (n-1)/k-1\rfloor}\alpha - (-1)^{\lfloor (n-1)/k-1\rfloor}\sum_{j=0}^{(n-1)\bmod k} L_j$$
$$-\sum_{j=2}^{\lfloor (n-1)/k\rfloor} 2(-1)^{j+\lfloor (n-1)/k\rfloor}\cdot\left(\sum_{m=0}^{\substack{((n-1)\bmod k)\\+(j-1)k}} L_m\right)$$

Proof of Lemma 1: The proof is derived from Lemma 8 in [33] by substituting L_i with $(L_i+\delta_i)$ for $i\in[0,n-k]$.

Lemma 1. *The ordering constraint $v_i < v_{i+1}$ can be expressed as a function of A) the offsets $\xi = (\xi_0, \dots, \xi_{k-1})$, B) the distortion of each Voronoi segment $\delta = (\delta_0, \dots, \delta_{n-k})$, and C) the lengths of a subset of Voronoi segments $L_0, \dots, L_{n-k}$. Specifically by using Lemma 8 from [33] we get the following cases:*

- *if $0 \le i < k-1$, then $v_i < v_{i+1}$ can be written as:*

$$-\xi_i + \xi_{i+1} - \delta_{i+1} < c_{i,i+1}, \text{ where } c_{i,i+1} = L_{i+1}$$

- *if $i = k-1$, then $v_i < v_{i+1}$ can be written as:*

$$-\xi_{k-1} - \xi_0 + \sum_{1 \le l \le k-1} \delta_l < c_{k-1,k}, \text{ where } c_{k-1,k} = -\sum_{1 \le l \le k-1} L_l$$

- *if $k \le i < 2k-1$, then $v_i < v_{i+1}$ can be written as:*

$$\xi_{i \bmod k} - \xi_{i \bmod k+1} - \delta_{i \bmod k+1} < c_{i,i+1}, \text{ where } c_{i,i+1} = L_{i \bmod k+1}$$

- *if $i = 2k-1$, then $v_i < v_{i+1}$ can be written as:*

$$\xi_{k-1} + \xi_0 - \delta_k - \sum_{1 \le l \le k} \delta_l < c_{2k-1,2k}, \text{ where } c_{2k-1,2k} = L_k + \sum_{1 \le l \le k} L_l$$

- *if $2k \le i < n-1$ and $(i+1) \bmod k \ne 0$, then $v_i < v_{i+1}$ can be written as:*

$$(-1)^{\lfloor i/k-1 \rfloor}(\xi_{i \bmod k} - \xi_{(i+1) \bmod k}) - (-1)^{\lfloor i/k-1 \rfloor}(\delta_{(i+1) \bmod k}) - \sum_{2 \le j \le \lfloor i/k \rfloor} (-1)^{j+\lfloor i/k \rfloor} 2(\delta_{i \bmod k+(j-1)k+1}) < c_{i,i+1}$$

$$, \text{ where } c_{i,i+1} = (-1)^{\lfloor i/k-1 \rfloor} L_{(i+1) \bmod k} + \sum_{2 \le j \le \lfloor i/k \rfloor} (-1)^{j+\lfloor i/k \rfloor} 2L_{i \bmod k+(j-1)k+1}$$

- *if $2k \le i < n-1$ and $(i+1) \bmod k = 0$, then $v_i < v_{i+1}$ can be written as:*

$$(-1)^{\lfloor i/k \rfloor+1}(\xi_{i \bmod k} + \xi_{(i+1) \bmod k}) - (-1)^{\lfloor i/k \rfloor+1}(\sum_{1 \le l \le k} \delta_l) - (-1)^{\lfloor i/k \rfloor+1}(\delta_k) - \sum_{2 \le j \le \lfloor i/k \rfloor} (-1)^{j+\lfloor i/k \rfloor} 2(\delta_{jk}) < c_{i,i+1}$$

$$, \text{ where } c_{i,i+1} = (-1)^{\lfloor i/k \rfloor+1} \left(\sum_{1 \le l \le k} L_l \right) + (-1)^{\lfloor i/k \rfloor+1} L_k + \sum_{2 \le j \le \lfloor i/k \rfloor} (-1)^{j+\lfloor i/k \rfloor} 2L_{jk}$$

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.

Lemma 2. *The boundary constraints $\alpha < v_0$ and $v_{n-1} < \beta$ can be expressed as a function of A) the offsets $\xi = (\xi_0, \dots, \xi_{k-1})$, B) the distortion of each Voronoi segment $\delta = (\delta_0, \dots, \delta_{n-k})$, and C) the lengths of a subset of Voronoi segments $L_0, \dots, L_{n-k}$. Specifically we have the following cases*

- *for the lower boundary*

$$\xi_0 - \delta_0 \le c_l, \text{ where } c_l = L_0$$

- *for the upper boundary*
  -if $k \le n-1 < 2k$:

$$\xi_{(n-1) \bmod k} + \sum_{j=0}^{n-k-1} \delta_i < c_u, \text{ where } c_u = L_{n-k}$$

-if $2k \le n-1$:

$$(-1)^{\lfloor (n-1)/k-1 \rfloor} \xi_{(n-1) \bmod k} + (-1)^{\lfloor (n-1)/k-1 \rfloor} \sum_{j=0}^{(n-1) \bmod k} \delta_j + \sum_{j=2}^{\lfloor (n-1)/k \rfloor} 2(-1)^{j+\lfloor (n-1)/k \rfloor} \cdot \left(\sum_{m=0}^{\substack{((n-1) \bmod k) \\ +(j-1)k}} \delta_m \right) < c_u$$

$$, \text{ where } c_u = \beta - (-1)^{\lfloor (n-1)/k-1 \rfloor} \alpha - (-1)^{\lfloor (n-1)/k-1 \rfloor} \sum_{j=0}^{(n-1) \bmod k} L_j - \sum_{j=2}^{\lfloor (n-1)/k \rfloor} 2(-1)^{j+\lfloor (n-1)/k \rfloor} \cdot \left(\sum_{m=0}^{\substack{((n-1) \bmod k) \\ +(j-1)k}} L_m \right)$$

Jackknife Estimators.

$$
\begin{aligned}
\hat{N}_{J(4)} =& d + \frac{4m-10}{m}f_1 - \frac{6m^2-36m+55}{(m-1)m}f_2 + \frac{4m^3-42m^2+148m-175}{m(m-1)(m-2)}f_3 - \frac{(m-4)^4}{(m-3)(m-2)(m-1)m}f_4\\
\hat{N}_{J(5)} =& d + \frac{5m-15}{m}f_1 - \frac{10m^2-70m+125}{m(m-1)}f_2 + \frac{10m^3-120m^2+485m-660}{m(m-1)(m-2)}f_3 - \frac{(m-4)^5-(m-5)^5}{m(m-1)(m-2)(m-3)}f_4 + \frac{(m-5)^5}{(m-4)(m-3)(m-2)(m-1)m}f_5\\
\hat{N}_{J(6)} =& d + \frac{6m-21}{m}f_1 - \frac{15m^2-120m+245}{m(m-1)}f_2 + \frac{20m^3-270m^2+1230m-1890}{(m-2)(m-1)m}f_3 - \frac{15m^4-300m^3+2265m^2-7650m+9751}{(m-3)(m-2)(m-1)m}f_4\\
&+ \frac{(m-5)^6-(m-6)^6}{(m-4)(m-3)(m-2)(m-1)m}f_5 - \frac{(m-6)^6}{(m-5)(m-4)(m-3)(m-2)(m-1)m}f_6\\
\hat{N}_{J(7)} =& d + \frac{7m-28}{m}f_1 + \frac{-21m^2+189m-434}{m(m-1)}f_2 + \frac{35m^3-525m^2+2660m-4550}{(m-2)(m-1)m}f_3 + \frac{-35m^4+770m^3-6405m^2+23870m-33621}{(m-3)(m-2)(m-1)m}f_4\\
&+ \frac{21m^5-630m^4+7595m^3-45990m^2+139867m-170898}{(m-4)(m-3)(m-2)(m-1)m}f_5 + \frac{(m-7)^7-(m-6)^7}{(m-5)(m-4)(m-3)(m-2)(m-1)m}f_6\\
&+ \frac{(m-7)^7}{(m-6)(m-5)(m-4)(m-3)(m-2)(m-1)m}f_7\\
\hat{N}_{J(8)} =& d + \frac{8m-36}{m}f_1 + \frac{-28m^2+280m-714}{(m-1)m}f_2 + \frac{56m^3-924m^2+5152m-9702}{(m-2)(m-1)m}f_3 + \frac{-70m^4+1680m^3-15260m^2+62160m-95781}{(m-3)(m-2)(m-1)m}f_4 +\\
&+ \frac{56m^5-1820m^4+23800m^3-156520m^2+517608m-688506}{(m-4)(m-3)(m-2)(m-1)m}f_5 + \frac{-28m^6+1176m^5-20650m^4+194949m^3-1029028m^2+2920008m-343615}{(m-5)(m-4)(m-3)(m-2)(m-1)m}f_6 +\\
&+ \frac{(m-7)^8-(m-8)^8}{(m-6)(m-5)(m-4)(m-3)(m-2)(m-1)m}f_7 + \frac{(-1)(m-8)^8}{(m-7)(m-6)(m-5)(m-4)(m-3)(m-2)(m-1)m}f_8\\
\hat{N}_{J(9)} =& d + \frac{9m-45}{m}f_1 + \frac{-36m^2+396m-1110}{(m-1)m}f_2 + \frac{84m^3-1512m^2+9198m-18900}{(m-2)(m-1)m}f_3 +\\
&+ \frac{(1/120)(m-9)^9-(1/24)(m-8)^9+(1/12)(m-7)^9-(1/12)(m-6)^9+(1/24)(m-5)^9-(1/120)(m-4)^9}{(m-3)(m-2)(m-1)m}f_4 +\\
&+ \frac{(1/24)(m-9)^9-(1/6)(m-8)^9+(1/4)(m-7)^9-(1/6)(m-6)^9+(1/24)(m-5)^9}{(m-4)(m-3)(m-2)(m-1)m}f_5 +\\
&+ \frac{(1/6)(m-9)^9-(1/2)(m-8)^9+(1/2)(m-7)^9-(1/6)(m-6)^9}{(m-5)(m-4)(m-3)(m-2)(m-1)m}f_6 + \frac{(1/2)(m-9)^9-(m-8)^9+(1/2)(m-7)^9}{(m-6)(m-5)(m-4)(m-3)(m-2)(m-1)m}f_7 +\\
&+ \frac{-(m-8)^9+(m-9)^9}{(m-7)(m-6)(m-5)(m-4)(m-3)(m-2)(m-1)m}f_8 + \frac{(m-9)^9}{(m-8)(m-7)(m-6)(m-5)(m-4)(m-3)(m-2)(m-1)m}f_9\\
\hat{N}_{J(10)} =& d + \frac{10m-55}{m}f_1 + \frac{-45m^2+540m-1650}{(m-1)m}f_2 + \frac{120m^3-2340m^2+15420m-34320}{(m-2)(m-1)m}f_3 + \frac{-210m^4+5880m^3-62370m^2+296940m-535227}{(m-3)(m-2)(m-1)m}f_4\\
&+ \frac{252m^5-9450m^4+142800m^3-1086750m^2+4164510m-6427575}{(m-4)(m-3)(m-2)(m-1)m}f_5\\
&+ \frac{-210m^6+10080m^5-202650m^4+2184000m^3-13306545m^2+43453200m-59411605}{(m-5)(m-4)(m-3)(m-2)(m-1)m}f_6\\
&+ \frac{120m^7-7140m^6+182700m^5-2606100m^4+22380120m^3-115700130m^2+333396850m-413066170}{(m-6)(m-5)(m-4)(m-3)(m-2)(m-1)m}f_7 + \frac{-(1/2)(m-10)^{10}+(m-9)^{10}-(1/2)(m-8)^{10}}{m(m-1)(m-2)(m-3)(m-4)(m-5)(m-6)(m-7)}f_8\\
&+ \frac{(m-9)^{10}-(m-10)^{10}}{m(m-1)(m-2)(m-3)(m-4)(m-5)(m-6)(m-7)(m-8)}f_9 + \frac{-(m-10)^{10}}{m(m-1)(m-2)(m-3)(m-4)(m-5)(m-6)(m-7)(m-8)(m-9)}f_{10}
\end{aligned}
$$