Differentially Private Access in Encrypted Search: Achieving Privacy at a Small Cost?

Daniel Pöllmann*
ETH Zurich
Zurich, Switzerland
dpoellmann@ethz.ch

Tianxin Tang†
Eindhoven University of Technology
Eindhoven, Netherlands
ac.tianxin.tang@gmail.com

Abstract

Encrypted search focuses on protecting sensitive data in outsourced environments while enabling private queries. Although standard encrypted search algorithms are efficient, they often leak some information about the queries and data. One such leakage is the access pattern on the outsourced storage. Recent leakage-abuse attacks have exploited this seemingly harmless leakage to successfully recover both queries and data, shifting research priorities towards finding the right balance between privacy and performance. While some proposals leverage oblivious RAM or other oblivious data structures to hide the access pattern, they typically incur significant bandwidth costs.

In response, researchers have developed new schemes that ensure access leakage satisfies *differential privacy (DP)*. Yet the security implications of these new guarantees remain unclear. Especially, compared with conventional differential privacy, the application and threat model are significantly different. To understand these implications, we investigate two concrete instances of (encrypted) range-query schemes (appeared in SODA '19 and CCS '22) that achieve differentially private access. We analyze their security guarantees using inference attacks to recover queries and data on real-world datasets. Our findings raise a critical concern that ensuring access leakage is differentially private either falls short of providing strong security for the queries and data, diverging from the initial goals, or offers only weak security but at a high efficiency/correctness cost. As part of our analysis, we also propose a generic security definition for DP access, and identify two general techniques for leakage mitigation, *bucketization* and *partitioning*, that may be of independent interest.

1 Introduction

Encrypted Search. Outsourcing data to cloud services has become increasingly popular due to the advantages of on-demand storage and high data availability. However, when sensitive data is involved, significant privacy concerns arise, particularly given that cloud services are vulnerable to breaches. While standard blockcipher-based symmetric encryption might seem like an option, this approach compromises essential search functionality. Consider a dynamic medical database: analysts might need to perform *point queries* for exact matches or *range queries* to find records with values within specific ranges. The process of downloading an encrypted database, decrypting it, conducting searches and updates, re-encrypting, and then re-uploading is simply impractical.

Specialized solutions to this challenge largely fall within the scope of *searchable symmetric encryption (SSE)* and *structured encryption (STE)*, which we will collectively refer to as *encrypted search algorithms (ESAs)*.[1] In ESAs, a *database* is commonly represented as a map (or multi-map)—a list of keyword-value (i.e., label-value) pairs. A client encrypts their database with a secret key and then transfers this encrypted database to a potentially untrusted server. Later, the client can query this encrypted database using the same secret key. To address real-world threats, such as server breaches and insider attacks, ESAs model the server as a passive-yet-persistent adversary capable of monitoring communication traffic. Under this threat model, ESAs aim to achieve both *query privacy*, protecting the keywords queried by the client, and *data privacy*, ensuring the confidentiality of the database's content.

Leakage. ESAs generally achieve high query efficiency by allowing some leakage. For instance, a passive-yet-persistent adversary (e.g., the server) can identify repeated queries based on recurring search tokens, known as the *query-equality pattern*; and the count and size of the encrypted records returned, termed the *volume pattern*. Alarmingly, this seemingly harmless leakage has been exploited in various leakage-abuse attacks against ESAs [8, 26, 32, 50, 57], challenging their security guarantees and practical utility.

ORAM and Overhead. Since the above-mentioned leakage typically stems from the access pattern to untrusted storage, oblivious RAM (ORAM) presents a potential solution. When treating untrusted storage as an array of memory blocks, ORAM protects a program's logical access pattern during operations. This security property is known as *(perfect) obliviousness*.

While ORAM does not directly solve the encrypted search problem, its obliviousness notion has inspired various oblivious data structures. Among these, oblivious maps have been used to construct ESAs that hide the (logical) access pattern, providing strong security [18, 19, 49].

Unfortunately, this *"obliviousness"* security gain comes with substantial costs. Research shows that the lower bound for *asymptotic computation* and *communication costs* when accessing a single block is logarithmic in N, the total number of memory blocks [44]. In practical terms, a theoretically efficient construction like PathORAM can introduce a 240-fold overhead [65],[2] making it unsuitable for many applications. Additional factors, such as the *number of roundtrips*, which can also scale logarithmically with N, remain unaddressed. If one tries to achieve *perfect volume-hiding* by padding each query

*The foundation of this research was laid in the master's thesis of Daniel Pöllmann, titled "Differential Obliviousness and its Limitations", at ETH Zurich.
†Most of this work was conducted while Tianxin Tang was affiliated with the Applied Cryptography Group at ETH Zurich.

[1] See Section 7 for related work.
[2] This overhead is for $N = 2^{30}$ and bucket size = 4.

response to its maximum size—the communication overhead might equate to that of transferring the entire database.[3]

Leakage Mitigation Problem. We now face a conundrum. Ideally, we want ESAs with strong security, preferably hiding the access pattern like ORAM. Meanwhile, we also aim to reduce volume leakage since research has shown that volume leakage alone can enable reconstruction of an encrypted database [26, 42]. Yet, it is certainly impractical to eliminate all types of leakage. The challenge then becomes: how can we build an efficient ESA with limited leakage that still provides reasonable security?

Recent research has incorporated *differential privacy (DP)* into both ESAs and ORAM constructions [2, 11, 14, 16, 52, 60, 62, 66], suggesting a potential avenue toward balancing privacy and efficiency. If we cannot efficiently suppress all access leakage, could making this leakage differentially private be the answer? Intriguingly, several studies suggest this approach incurs only minor efficiency overhead [11, 14].[4] However, there is an important caveat. Conventional *differential privacy (DP)* is typically employed for **statistical disclosure of publicly released datasets** to provide *individual membership privacy*. It ensures that disclosed statistics do not vary significantly regardless of individual data inclusion. This application setting differs fundamentally from that of ESAs and ORAMs, which focus on protecting the semantic privacy of queries and data; ideally, leak nothing about either. This distinction also implies significantly different models. Although DP is well-established as a privacy notion, its application in leakage mitigation for ESAs or ORAMs is often adopted without comprehensive justification. This raises critical concerns about suitable applications, threat models, and practical privacy guarantees of proposed constructions, as in [11, 16]. Our work aims to bridge this gap by addressing the following question:

Does differentially private access achieve (semantic) query and data privacy at a small cost?

Unfortunately, our answer is negative. For adversaries capable of observing the transcripts of differentially private access (without further privacy guarantees), we show that the mechanism alone cannot provide sufficient semantic *query* and *data privacy*, particularly in encrypted search. This is supported by concrete counterexamples examining two range-query structures: DO-Range (SODA '19) [11, 12] and ϵ-OPE-Range (CCS '22) [16],[5] which we will discuss later. We can approximately recover the *normalized histogram* of the plaintext data,[6] without access to any auxiliary distribution, typically assumed in leakage-abuse attacks [29, 51]. For both DO-Range and ϵ-OPE-Range, we need to observe sufficient queries for accurate recovery. We argue that these counterexamples are sufficient to demonstrate that such access mechanisms fall short of the standard semantic security needed for encrypted search.

Even with more relaxed differential privacy for individual membership privacy, we show that there is a hefty price tag in utility, contradicting the small cost claim made in [11]. The marginal improvement in privacy regarding individual membership deviates from the original objective.

Differential Privacy. Before delving into our analysis, let us first introduce *differentially private access leakage (DP access leakage)*. This requires some background knowledge of *differential privacy (DP)*.

Differential privacy is a notion centered on *individual membership privacy*. DP is typically achieved using a differentially private mechanism. This mechanism guarantees that, when running statistical queries on the collected data, the output will not significantly change based on any single individual's input.

To illustrate the DP mechanism, consider a simple medical database containing patients' records. This database has only two boolean attributes: being a smoker and having lung cancer. An analyst wants to compute the *count* of patients with lung cancer. For this count query, instead of returning the exact count, a randomized mechanism $\mathcal{M}$ is used. This mechanism, $\mathcal{M}$, adds random noise to the count, drawn from a (discrete) Laplace distribution, for example, by adding 1. We can then show that $\mathcal{M}$ satisfies DP. Intuitively, this is straightforward, as an individual's input contributes at most one to the count output with high probability, regardless of its value.

More specifically, $\mathcal{M}$ ensures ϵ-DP. Here, ϵ bounds the probability that an arbitrary adversary that might distinguish between the $\mathcal{M}$'s outputs from *neighboring databases* that differ by just one individual input.[7] This parameter ϵ, commonly known as the *privacy budget*, indicates the leakage from each query using $\mathcal{M}$. To ensure individual membership privacy, it is critical to restrict the number of chosen queries based on ϵ. Without this limitation, an adversary, after observing multiple output samples, might recover the actual count, thereby compromising individual membership privacy.

Differentially Private Access. Recall that, in ESAs, access leakage is typically the primary concern. Similarly, ORAM-related oblivious constructions protect against this access leakage. To reduce access leakage, while not making it entirely oblivious for efficiency reasons, one might consider defining differentially private access leakage. This would require introducing a DP access mechanism for queries. However, a key distinction is that, in conventional DP, $\mathcal{M}$ **intrinsically encapsulates the queries**, particularly the query type, such as *count* and *average*. Conversely, in the contexts of ESAs and ORAM, **the DP access mechanism must take as input the query message**, for example, a range $[s, e]$ or a point x. It is crucial to emphasize that, in different applications, the useful security notions and threat models may differ. We discuss these aspects in Sections 3 and 5.

Our Analysis on DO-Range **and** ϵ-OPE-Range**.** What exactly is guaranteed with DP access? As case studies, we closely examine two *range-query* constructions: DO-Range (SODA '19) [11, 12] and ϵ-OPE-Range (CCS '22) [16], analyzing their DP definitions, techniques, correctness, and efficiency. Our analysis reveals fundamental differences in techniques for achieving DP in access compared with standard DP. We select DO-Range because it serves as a theoretical foundation for DP access (i.e., differential obliviousness)

[3] To illustrate, suppose a frequent keyword matches every document in the database. For perfect volume-hiding, the entire database would need to be returned when searching for a non-frequent keyword.

[4] In [11], for the proposed DO-Range, the asymptotic computation overhead for encrypted range search compared with the unencrypted baseline is only $O(\log \log N)$ for insertion, and no asymptotic increase in searches.

[5] DO-Range only claims data privacy but not query privacy.

[6] The y coordinates indicate the fractions of the total elements. We use *histogram* interchangeably with *normalized histogram* in later discussions

[7] This interpretation is based on the ϵ-indistinguishability notion first introduced in the seminal work [22].

and has a concrete range-query construction. This work is followed by a line of research, for more advanced DP access composition theorems [69], and possible database applications [58, 59, 68]. We choose ϵ-OPE-Range as it aims to strengthen order-preserving encryption (OPE) [3] used for encrypted range queries. Both range-query schemes can achieve point queries at the same time by setting the range to 1. The *bucketization* and *partitioning* techniques that underline the two schemes are general and have been used in leakage mitigation for ESAs such as [19].

Surprisingly, these two aforementioned techniques in DP access shift the perspective on the *privacy budget* and its limit on the number of queries. In addition, setting the failure probability δ to prevent full recovery of individual data leads to significant efficiency overhead. These aspects are fundamentally different from the traditional DP notion. Because the DP access notion is hard to interpret by itself, we use *inference attack* (i.e., leakage-abuse attacks) on the data and queries to demonstrate the concrete privacy guarantees. We evaluate their privacy by running data recovery experiments on real-world datasets encrypted using the two schemes. The goal is to evaluate privacy guarantees and efficiency costs on different privacy parameters. We found changing the viewpoint on privacy reveals that a high-recovery rate on the (normalized) histogram does not violate the DP access notion. However, the marginal privacy gain, at the expense of efficiency and correctness, raises doubts about their practical usefulness.

Our Contributions. In this work, *we contrast differentially private leakage in access with standard differential privacy used in data analytics*, examining their applications, threat models, definitions, and techniques. While differential privacy is well-established, its application to private data access and associated threat models remains unclear. We specifically highlight the limitations of differentially private access in encrypted search.

We formalize differentially private access using standard structured encryption (STE) syntax and show that it can be instantiated by the two range-query schemes we analyze, providing security proofs that connect the relevant security notions. Since privacy parameters (ϵ and δ) are difficult to interpret, we conduct inference attacks on queries and data using real-world datasets to evaluate semantic security. Our analysis reveals that these two schemes fail to provide strong semantic security despite not violating differential privacy. In addition, our findings suggest that achieving differential privacy (i.e., individual membership privacy) with high correctness imposes significant efficiency costs.

As differential privacy gains increasing adoption in encrypted search, as evidenced in studies such as [2, 58, 59, 68], our work highlights the need to critically re-evaluate differentially private access as an effective leakage mitigation strategy.

Of particular interest, our inference attacks do not require any auxiliary data distribution. This certainly does not preclude other attack strategies but suffices to advance our understanding of access leakage and raises caution for practitioners during deployment.

2 Preliminaries

Notation and Convention. For any $n \in \mathbb{N}$, we let $[n]$ denote the discrete range $[1, n]$. The algorithms are randomized and polynomial-time (in the security parameter) unless otherwise specified. We use out $\leftarrow\!\$\ A$ to indicate out is generated by some algorithm A using its internal randomness. We use PPT for the abbreviation of probabilistic-polynomial-time and let $\mathsf{negl}(\cdot)$ denote the negligible function.

Interactive Algorithm. For an arbitrary (two-party) interactive algorithm $\mathcal{I}$ run between A and B, we use the notation $(\mathsf{output}_A, \mathsf{output}_B) \leftarrow [\mathcal{I}_A(\mathsf{input}_A), \mathcal{I}_B(\mathsf{input}_B)]$.

2.1 Differential Privacy (DP)

Overview. We start with a brief overview of differential privacy (DP). Dwork et al. [22] introduced DP to capture formally *individual membership privacy*. Specifically, it is designed for scenarios where there is a need to learn statistical information about a population without compromising the information of individual membership. The seminal work [22] introduced a widely used definition based on indistinguishability. This notion offers a worst-case guarantee of individual membership privacy: even when an adversary chooses two databases that differ by only one record (i.e., data from a single individual), and runs the same query through a DP-mechanism $\mathcal{M}$ on both databases separately, it can only distinguish the two query outputs with a small probability.

This DP-mechanism $\mathcal{M}$ encapsulates the query functionality. For example, it can be an aggregate function such as *count* and *average*. To construct $\mathcal{M}$ for a fixed function, Dwork et al. [22] propose a general technique by adding *random noise* to the function's output.

More formally, within the DP definition, there exists a *privacy budget* $\epsilon > 0$. To offer flexibility in construction, a relaxed DP notion incorporates *failure probability* $\delta \geq 0$. To avoid catastrophic privacy breaches, δ should be set to a value negligible in the number of records in the database.

Definition 1 (Differential Privacy ((ϵ, δ)-DP) [23])**.** A randomized mechanism $\mathcal{M}$ is (ϵ, δ)-DP if, for all neighboring databases D_0, D_1, every output set $S \subseteq \mathrm{Range}(\mathcal{M})$,

$$\Pr(\mathcal{M}(D_0) \in S) \leq e^{\epsilon} \cdot \Pr(\mathcal{M}(D_1) \in S) + \delta,$$

where the probabilities are over the randomness used by $\mathcal{M}$.

ϵ-DP**.** We also use ϵ-DP to denote (ϵ, δ)-DP for $\delta = 0$.

Laplace Mechanism & Composition Theorem. We briefly review the Laplace mechanism that achieves ϵ-DP [23] and the standard DP composition theorem in Appendix A.

3 Applications

As mentioned in the introduction, DP notions are widely adopted, making it easier to motivate the notion of DP access. However, the potential applications, threat models, and privacy guarantees are difficult to interpret in work such as [11, 12, 16]. To pave the way for our security analysis, we first summarize real-world applications requiring query/data privacy in databases.

Statistical Disclosure. Government agencies are often obliged to publicly release certain data. Due to concerns about individual privacy, data is sanitized before its release. For example, the US Bureau collects demographic data used for allocating electoral votes, and sanitizes it before release to enable analysis while preserving anonymity.

Private Data Analytics. Companies like Google and Apple collect user data to improve services and recommendations. This collected user data, stored either locally or in the cloud, should be protected against unauthorized access, and support data analytics while preserving individual privacy.

Encrypted Databases / ESAs. Database vendors now widely offer encryption-at-rest. There is a growing demand for more advanced features like searching on encrypted data. This is illustrated by queryable encryption, a database extension by major vendor MongoDB that supports database encryption with secure queries.[8]

Trusted Execution. For trusted execution environments (TEEs) with limited secure storage, secure mechanisms are needed to access larger, untrusted storage.

3.1 Entities

For the applications mentioned before, we consider the following entities: *individual:* produces data, contributing one record to the database. *Sanitizer:* sanitizes the raw data (e.g., service providers, government agencies). *Data owner:* owns the original database (e.g., service providers, government agencies). *Analyst:* studies the (sanitized) database.

Client and Server. We use *client* to refer to entities with the secret key for encryption and decryption. We use *server* to refer to an untrusted storage service who does not have the secret key, but handles data access requests.

3.2 Database Functionality and Privacy

Functionality. The applications we have discussed require a database to support basic search operations (e.g., *point queries* and *range queries*) and *aggregate functions* such as *count* and *average*.

Privacy. We provide a high-level overview of the privacy notions suitable for these applications. When collecting individual data for analytics, it is vital to preserve *individual membership privacy*, even when adversaries have access to query outputs. This assurance increases an individual's willingness to share their data. Such a privacy requirement perfectly aligns with *differential privacy*.

When storing a private database after encryption, either on a local server or in the cloud, it is essential to protect against data breaches. By modeling the server as a passive-yet-persistent adversary, capable of observing the transcripts used to query the encrypted database, we can simulate real-world threats, including data breaches. Within this threat model, preserving the content of the private database is known as *data privacy*. Additionally, if queries reveal more information than query types, then this information should also be protected, referred to as *query privacy*. Protecting both query and data privacy is the focus of encrypted search. We use the generalized notion *structured encryption* (STE) and its adaptive security notion. We recall the STE syntax and security definition in Appendix B and discuss the threat models used for our analysis in Section 5.

[8] https://www.mongodb.com/docs/manual/core/queryable-encryption

4 Differentially Private Access

We formalize the notion of differentially private access used in our analysis. Contrary to conventional indistinguishability, we require the transcripts (and encrypted data structure), which includes the physical access pattern, to be differentially private. We provide an adaptive DP-Access notion here, and a non-adaptive one in the full version of this paper.

Note that we only consider Find queries that do not modify the database in our definition. If insertion queries are allowed, one way to address this is to enforce that the insertion queries are neighboring, only allowing modification of one label-value pair in the data structure, similar to [11, 12]. However, using this strategy introduces restrictions to the DP composition theorem shown in [70]. Instead, our notion is designed to address DP access leakage without modifying the data structure. This notion resembles the conventional DP notion, with respect to taking the neighboring databases as input into the DP mechanism (i.e., the setup phase). We show that our DP access notion is natural and sufficient to capture the access leakage analyzed in our study. We also provide proofs in the full version demonstrating that the two range query structures we analyze, which satisfy their respective DP security notions, also satisfy our DP-Access notions.

To define DP-Access, we start with defining the neighboring condition of the data structures.

Definition 2 (Neighboring Data Structures)**.** Given label space $\mathbb{L}$, value space $\mathbb{V}$, and operation type space $\mathbb{O}$, consider a data structure $\mathsf{ds} \subseteq \mathbb{L} \times \mathbb{V}$ with functionality $\mathcal{F}$. Let $\mathtt{dist}$ be an abstract distance function over $\mathbb{L} \times \mathbb{V}$. We say that ds_0 and ds_1 are neighboring if and only if they are of equal size and $\mathtt{dist}(\mathsf{ds}_0, \mathsf{ds}_1) = 1$.

Definition 3 (Adaptive Differentially Private Access)**.** Given label space $\mathbb{L}$, value space $\mathbb{V}$, and operation type space $\mathbb{O}$, let Π be an STE protocol with functionality $\mathcal{F}$. Let $\lambda \in \mathbb{N}$ be the security parameter. For $b \in \{0, 1\}$, we define the security experiment in Figure 1.

$\mathbf{Exp}^{\text{dp-access},b}_{\Pi,\mathcal{A},s}(1^\lambda, z, \mathsf{ds}^0, \mathsf{ds}^1)$:

1: $(\mathsf{sk}, \mathsf{eds}_1) \leftarrow\$ \mathsf{Setup}(1^\lambda, z, \mathsf{ds}^b)$.
2: $\mathcal{A}$ is given ds^0, ds^1, λ, z and eds_1.
3: $\mathcal{A}$ adaptively makes s queries in any order,
4: **for** all $i \in [s]$ **do**
5: Access($\mathtt{Find}, \ell_i, \ell'_i, \perp$):
6: where $\ell_i, \ell'_i \in \mathbb{L}$,
7: returns to $\mathcal{A}$ the transcript of honest execution of
8: $[\mathsf{Access}_{\mathbf{C}}(\mathsf{sk}, \mathtt{Find}, \ell_i, \ell'_i, \perp), \mathsf{Access}_{\mathbf{S}}(\mathsf{eds}_i)]$ and
9: Server $\mathbf{S}$'s output eds_{i+1}.
10: **end for**
11: Finally, $\mathcal{A}$ outputs bit b'.
12: The experiment returns the same bit b'.

Figure 1: Adaptive Differentially Private Access (DP-Access**) Experiment.**

We say that Π has adaptive (ϵ, δ)-DP access leakage if for all $z \in \{0,1\}^*$, all neighboring ds_0 and ds_1 over $\mathbb{L} \times \mathbb{V}$, all s which is a polynomial (in λ), all PPT adversary $\mathcal{A}$, all $b' \in \{0,1\}$,

$$\Pr\left[\mathbf{Exp}^{\text{dp-access},b}_{\Pi,\mathcal{A},s}(1^\lambda, z, \mathsf{ds}^0, \mathsf{ds}^1) = b'\right] \leq$$
$$e^\epsilon \cdot \Pr\left[\mathbf{Exp}^{\text{dp-access},1-b}_{\Pi,\mathcal{A},s}(1^\lambda, z, \mathsf{ds}^0, \mathsf{ds}^1) = b'\right] + \delta.$$

Remark 1 (On Failure Probability δ). Similar to the conventional DP notion, to ensure individual privacy, the failure probability δ should be chosen as a negligible function in the number of label-value pairs in the data structure.

Remark 2 (On Symmetry). In the standard computational indistinguishability game, the equality relation applies to both sides. Similarly, the DP indistinguishability-based inequality also applies to both sides since the neighboring property is symmetric. We write the experiment bit b and $\mathcal{A}$'s output b' explicitly for the convenience of our proofs.

Remark 3 (On DP Volume Pattern). We use the term *DP access leakage* instead of *DP access pattern* for generality. Our notion is sufficient to capture the DP volume pattern proposed in [52]. Their work, however, lacks an analysis of the security guarantees. We discuss its follow-up cryptanalysis in related work (Section 7).

Using this Notion. As we look ahead to our analysis in Section 6, our DP access notions apply to both the range-query structures DO-Range and ϵ-OPE-Range that we study. We do not intend to show the equality of these security notions but rather to show the generality of our notion, capturing the essence of differentially private access in a private setting and making it suitable for our inference attacks.

The neighboring condition in our DP-Access notions incorporates an abstract distance function. We make the following instantiations.

For DO-Range, we let $\mathtt{dist}(\mathsf{ds}_0, \mathsf{ds}_1) = 1$ if and only if ds_0 and ds_1 differ in exactly one pair of label and value. We show that DO-Range, satisfying its adaptive differential obliviousness notion, also satisfies our adaptive DP-Access notion.

For ϵ-OPE-Range, we let $\mathtt{dist}(\mathsf{ds}_0, \mathsf{ds}_1) = 1$ if and only if ds_0 and ds_1 differ in exactly one pair of label and value, and their difference in the label pairs is exactly 1 under the ℓ_1 distance metric, the same as defined in DO-Range. We show that ϵ-OPE-Range, satisfying its ϵ-IND-FA-OCPA notion, also satisfies our non-adaptive DP-Access notion (Definition 6). Due to space limitations, we provide the proofs in the Appendix C.2 and C.3.

5 Threat Models

Although DP-Access seems a reasonable DP notion, the implications of the inherent DP access leakage are unclear. This is concerning because, despite many STE schemes having provable security with certain leakage functions, leakage-abuse attacks still apply. Hence, it is crucial to study its privacy implications and provide guidance for practitioners. This goal requires establishing suitable threat models. In this section, we discuss suitable threat models, referring to the potential applications and entities involved, as summarized in Section 3. Table 1 offers an overview, and we provide our justifications in the text. We note that statistical disclosure falls under the category of DP analytics. Private data analytics belongs to both DP analytics and encrypted databases (i.e., encrypted search). We use "EDB" as an abbreviation for "encrypted database", and use "encrypted database" and "encrypted data structure" interchangeably. In some schemes, such as DO-Range, there is no Setup phase; instead, the encrypted database EDB can be reconstructed using only the transcripts.

	DP Analytics	ESAs & DP Access
Entities	Data owner Analyst Sanitizer	Data owner Analyst
Adversarial Access	Query Query result	Transcript (EDB)
Adversarial Goal	Individual membership	Query Data

Table 1: Threat Models

Differentially Private Analytics (DP Analytics). The data owner possesses the database records and grants access to the database to an analyst through a sanitizer. The sanitizer responds to any statistical query from the analyst by adding noise to the query result, ensuring differential privacy.

Encrypted Databases (ESAs) / Differentially Private Access (DP Access). In this scenario, the data owner and analyst are the same entity, with the records being encrypted using their secret key. In ESAs, the analyst (i.e., client) accesses the encrypted database using the specified query algorithm. In DP access, the analyst accesses the (encrypted) database through a differentially private mechanism. This mechanism ensures that the access leakage satisfies differential privacy.

Formalization. To be precise, we formalize the threat models and adversarial goals, referencing our DP-Access notion. Given a label space $\mathbb{L}$ and a value space $\mathbb{V}$, we consider a database or data structure $\mathsf{ds} \subseteq \mathbb{L} \times \mathbb{V}$. The label space $\mathbb{L}$ is the domain of the searchable fields, such as numerical values (e.g., ages) in range queries. The value space $\mathbb{V}$ is non-searchable and associated with the searchable labels.

For a range-query scheme proven to satisfy DP-Access, our aim is to recover the labels. Our recovery goal aligns with almost all related work in the field of leakage-abuse attacks (See Section 7), specifically excluding the recovery of the non-searchable fields.

We illustrate later that the DP notion in private access intuitively provides a privacy guarantee on individual records, so it is reasonable to consider other recovery goals. In this work, we focus on recovering the normalized histogram of an encrypted database.

(Normalized) Histogram Recovery. Let us instantiate $\mathbb{L}$ as a finite discrete set with total ordering (e.g., numerical values). Let $\mathsf{ds} = (\ell_j, v_j)_{j\in[s]}$ denote the plaintext database, where $\ell_j \in \mathbb{L}$ and $v_j \in \mathbb{V}$. We now define a *histogram*. Given label $x \in \mathbb{L}$, let h_x denote the number of labels in the database equal to x, $h_x := \sum_{j\in[s]} \mathbb{1}\{\ell_j = x\}$. Then a histogram of ds can be defined as a list of h_x, $\mathtt{histogram} := (h_x)_{x\in\mathbb{L}}$. The normalized version can be defined as follows, $\overline{\mathtt{histogram}} := \left(\frac{h_x}{s}\right)_{x\in\mathbb{L}}$. To evaluate the

recovery accuracy of the normalized histogram, we define an error term: $\texttt{error} := \sum_{x \in \mathbb{L}} \frac{\left|\hat{h_x} - h_x\right|}{s}$, where $\hat{h_x}$ is the recovery.

We also provide a definition for the *approximate normalized histogram*. Given a partition set $\{p_i\}_{i \in [t]}$ over $\mathbb{L}$, let h_i denote the number of labels in the database that are in partition p_i, defined as $h_i := \sum_{j \in [s]} \mathbb{1}\{\ell_j \in p_i\}$. Then an approximate histogram of ds can be defined as a list of h_i, $\texttt{approxhist} := (h_i)_{i \in [t]}$. The normalized version is $\overline{\texttt{approxhist}} := \left(\frac{h_i}{s}\right)_{i \in [t]}$.

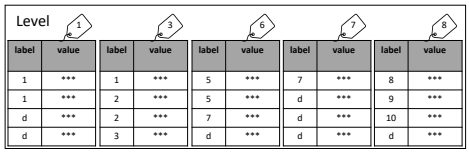


Figure 2: Example of buckets in a level. The interior points of the buckets are shown as tags. Dummy records are denoted with d. Encrypted record values are denoted by *.**

6 Analysis of DO-Range and ϵ-OPE-Range

Overview. In this section, we closely examine two range-query constructions: DO-Range (SODA '19) [11, 12] and ϵ-OPE-Range (CCS '22) [16]. Our analysis explores their DP access definitions, techniques, correctness, and efficiency.

These two schemes rely on different techniques to achieve DP access: we defer the comparison of the bucketization technique from DO-Range and the partitioning technique from ϵ-OPE-Range in Section 6.4.

Despite the wide adoption of DP and extensive research on auditing DP [45] and exploring membership inference attacks [15, 31], **its security risks in private data access remain unclear**, particularly in the potential applications outlined in Section 3. This uncertainty is concerning, given our focus on DO-Range and ϵ-OPE-Range. DO-Range establishes the theoretical foundations for DP access, as part of various algorithms and data structures, pushing forward the notion [58]; ϵ-OPE-Range promotes DP access as an effective privacy-enhancing approach to address the inherent security weakness in order-preserving encryption (OPE). Furthermore, the DP access definition with ϵ and δ parameters complicates understanding their query and data privacy guarantees. These motivate us to empirically evaluate their privacy guarantees using *inference attacks*, a standard approach in encrypted search [50]. This critical approach benefits the development of new and stronger constructions.

In essence, inference attacks simulate real-world attacks by evaluating the privacy risk through experiments. These experiments, with privacy risks quantified by query and data recovery rates of data encrypted with the suggested construction, are conducted under specific threat models. Such models may include, for example, access to transcripts or snapshots of encrypted data. We specify these threat models in Section 5. Notably, we highlight that we do not need any auxiliary distribution that is typically assumed in many inference attacks [26, 27, 50]. Additionally, we do not require strict attack assumptions, such as the requirement that each label must be queried at least once for our approximate recovery. These findings further challenge the usefulness of the DP access notion: it does guarantee DP, but falls short of providing sufficient privacy in many applications related to encrypted search.

6.1 Experimental Evaluation

For our inference attacks, we aim to approximately reconstruct the normalized histograms of the labels. We conducted our evaluation on two real-world datasets: the Actors IMDB Dataset for age information, and the Atlanta Employee Salary Dataset for both age and salary information. Further, we created a synthetic dataset for evaluation on databases containing a large number of records, specifically 2 million. We provide more details in Appendix D.3.

6.2 Construction and Attack on DO-Range

Overview of DO-Range (SODA '19) [11, 12]. As previously mentioned, DO-Range is a range-query structure that illustrates the practical relevance of the differential oblivious notion (i.e., DP access) [11, 12]. The computational overhead of DO-Range, compared with the unencrypted baseline, is only $O(\log \log N)$ for insertions, and it does not incur any asymptotic overhead in range queries. Furthermore, DO-Range does not use random noise (i.e., Laplace noise), but instead relies on a simple bucketization technique for DP access. Since DO-Range does not claim query privacy [11, 12], our analysis primarily focuses on its data privacy.

To help explain DO-Range, we start with the operations on plaintexts, introducing its use of encryption later. DO-Range supports dynamic insertion and range queries. Since we only utilize access leakage from range queries in our analysis, we omit the insertion algorithm and refer to [11, 12] for details.

We first briefly describe how DO-Range utilizes bucketization. Records are stored in buckets on the server, as shown in Figure 2. These buckets contain real records and are padded to a fixed bucket size with dummy records. Server storage is divided into multiple levels, each capable of holding multiple buckets. Further, each bucket is tagged with some value that lies between its *minimal* and *maximal* real record labels, as illustrated in Figure 2. This value is called an *interior point* of the bucket. The buckets are sorted based on these values in each level, which allows the server to efficiently find buckets corresponding to client queries.

The privacy guarantee of DO-Range is based on the fact that an adversary cannot determine *the number of real records* in any given bucket, nor can they *distinguish between real and dummy records*. Intuitively, if the buckets are sufficiently large and there is some uncertainty for the adversary regarding their contents, it is difficult for the adversary to know if a record is part of the database, aligning with the DP notion. We further elaborate on how these uncertainties contribute to privacy in the following discussion.

Bucketization. Bucketization refers to placing records into fixed-size buckets. Initially, the records are sorted by their labels in ascending order. A bucket is created by first sampling a random load r and inserting the r records with the smallest labels into the bucket. This bucket is then padded to the bucket size Z with dummy records, and the process continues for the next bucket.

Encryption. DO-Range encrypts every bucket (containing dummy and real records) using a standard blockcipher-based encryption scheme under the client's secret key. Interior points are encrypted for interactive mode and kept in plaintext for non-interactive mode, which we explain in detail later.

Interior Points and Sortedness. Recall that each bucket is tagged with an interior point, which is a value between the minimal and maximal label of the real records in the bucket. The interior point is randomly selected using an interior point mechanism, a differentially oblivious variant of [7]. Importantly, this value does not have to be one of the real record labels in the bucket.

After the client creates an (encrypted) bucket, the (encrypted) interior point is sent to the server along with the bucket. At each level, buckets are ordered by their interior points, and the records within these buckets are sorted by their labels. Further, due to this ordering, the labels of the records in a bucket always fall within the range defined by the interior point of the preceding bucket and that of the subsequent bucket.

(Non)-Interactive Modes. DO-Range operates in both interactive and non-interactive modes.

In the *interactive mode*, the interior points of buckets are encrypted and stored on the server such that the server cannot access them and learns no information about the labels of records in buckets. This mode increases the number of roundtrips for queries, as the client must iteratively query the server (e.g., through binary search) for interior points until it finds the buckets matching its find query.

In the *non-interactive mode*, the interior points of buckets are stored in plaintext on the server. This allows the server to use this information to identify the buckets that match a client's find query and return them directly. This mode requires only one roundtrip, thus improving performance.

Intuitively, the interactive mode provides better privacy. However, we show later that the interior point information can be approximated in the interactive setting by observing find queries.

Range Queries. Recall that each bucket is part of a level, and each level can hold multiple buckets. Within a level, buckets are sorted by their interior points. To make a find query, the client queries each level individually, as there is no relation among the levels. The client retrieves all matching buckets and filters out any false positives locally.

Attack Assumptions. We assume the attacker knows the minimal and maximal labels of the dataset and has access to the encrypted database and the transcripts of find queries. Our attack only exploits the access leakage from find queries, matching our DP-Access notion described in Section 4. To avoid insertion leakage, we assume the records are randomly shuffled before insertion. For high data recovery, sufficient queries must be observed. However, we relax the assumptions that all labels need to be queried and that the dataset must be dense,[9] as required for exact recovery in prior work such as [43], [28] and [37]. We note that [41] also does not require such assumptions; however, they only demonstrated the effectiveness of their approach on simulated data following the permuted-beta distribution.

[9]All labels need to appear in the dataset.

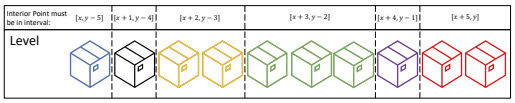


Figure 3: Multiple buckets are stored in a level, and we show the minimal sets as recovered by find queries. By knowing the minimal (x) and maximal (y) labels, it is possible to establish lower and upper bounds on the interior points of each bucket.

Attack Description. The core of our attack is simple yet effective. With sufficient queries, these queries enforce constraints on the interior points of the encrypted buckets. Using these constraints, we can estimate the data stored in these buckets. Our attack does not rely on strict assumptions about the data or the queries. Thus, it can be seen as an exploration of relaxing the attack assumptions in existing leakage-abuse techniques for approximate recovery. This attack is generic and also applicable to other schemes using the bucketization technique.

When clients make find queries, the attacker learns the encrypted buckets returned to the client as part of the transcript. For point queries, the attacker learns that each of the returned encrypted buckets must have the same interior point. Similarly, for range queries, all the returned buckets must have interior points that lie within the query's label range. This information partitions a level into sets, with the knowledge that buckets in distinct sets cannot possibly have the same interior points. These sets are referred to as *minimal sets*, as illustrated in Figure 3.

In addition to knowing that buckets are ordered by their interior points in each level, the information of minimal sets allows us to establish bounds on the interior points of the buckets. Therefore, we can constrain the possible values of the record labels within those buckets. For example, the first minimal set of buckets (ordered by ascending interior points) must have an interior point of at least the minimal label. The second minimal set must have a larger interior point; otherwise, it would have been part of the first minimal set. Analogously, we can bound the interior points in reverse order by starting at the last minimal set and the maximal label.

With this attack, we can recover the approximate interior points of any bucket, thereby constraining the labels of records that can be included in a bucket. Note that we cannot recover the exact number of real records in a bucket, as the bucket is padded with dummy records. However, we can estimate this value using the expected value of the randomized bucket load used by DO-Range.

In Algorithm 1, we present the pseudocode for our data (histogram) recovery attack on the buckets encrypted by DO-Range at a single level. The ATTACK procedure takes as input a list of query responses `rs`, the encrypted `buckets` stored on the server, and the label space $\mathbb{L}$. Specifically, we use $\texttt{rs}[i]$ to denote the response of the i-th query. If it returns a set of encrypted buckets `b`, we use the notation $\texttt{rs}[i] = \texttt{b}$.

In Algorithm 2, the CREATEMINSETS procedure takes the responses `rs` and the `buckets` and iterates over the buckets (Line 15) comparing the set of queries that returned the current bucket to the set of queries that returned the last bucket (Line 16). If both buckets

were part of exactly the same query responses, then the minimal set is extended with the current bucket (Line 17); otherwise, a new minimal set begins (Line 19).

The UpStep procedure computes the label bounds on the interior point of each bucket by iterating over the previously computed minimal sets (Line 10) and inferring for each of the contained buckets a lower bound on the interior point (Line 13). This lower bound starts with the minimal label (Line 9) and is updated to the next smallest label in each iteration (Line 16).

We omit the description of the DownStep procedure as it is nearly identical to the UpStep procedure, with the only difference being that the minimal sets are processed in reverse order to assign upper bounds for interior points. This attack procedure, runs in $O(|\texttt{buckets}| \cdot |\texttt{rs}|)$ time, where $|\texttt{buckets}|$ is the number of buckets and $|\texttt{rs}|$ is the number of queries.

Algorithm 1 DO-Range Attack

```
1:  procedure Attack(rs, buckets, 𝕃)
2:      labels_≥ ← {}
3:      minSets ← CreateMinSets(rs, buckets)
4:      UpStep(minSets, 𝕃)
5:      DownStep(minSets, 𝕃)
6:  end procedure
7:
8:  procedure UpStep(minSets, 𝕃)
9:      c_L ← 𝕃.min()
10:     for i = 1 to |minSets| do
11:         m ← minSets[i]
12:         for j = 1 to |m| do
13:             The interior point of bucket m[j] must
14:             be larger or equal c_L.
15:         end for
16:         c_L ← 𝕃.next()
17:     end for
18: end procedure
```

Results on Data Recovery. We now present our data (normalized histogram) recovery results for DO-Range, evaluated on the IMDB Age dataset and the Atlanta dataset. We selected a bucket size of 16 to achieve good utility, which we discuss later in the utility/privacy tradeoff part. We further conducted experiments with bucket size 512 on a larger synthetic dataset as shown in Figure 11 in Appendix D.1.

In Figure 4, we show how the access leakage from different percentages of labels appearing in the queries affects the recovery. A clear trend can be observed: the histogram shifts towards the ground truth as the percentage increases. At 70%, we achieve a good approximate recovery. This is further demonstrated by the `error` term, computed based on the accumulated distance from the ground truth, as defined in Section 5.[10] Indeed, this finding provides evidence that for approximate recovery, we can relax the strict attack assumptions of prior work, such as requiring every label to be queried.

[10]Since we cannot know how many dummy records are in a bucket, this recovery cannot be perfect.

Algorithm 2 Create Minimal Sets

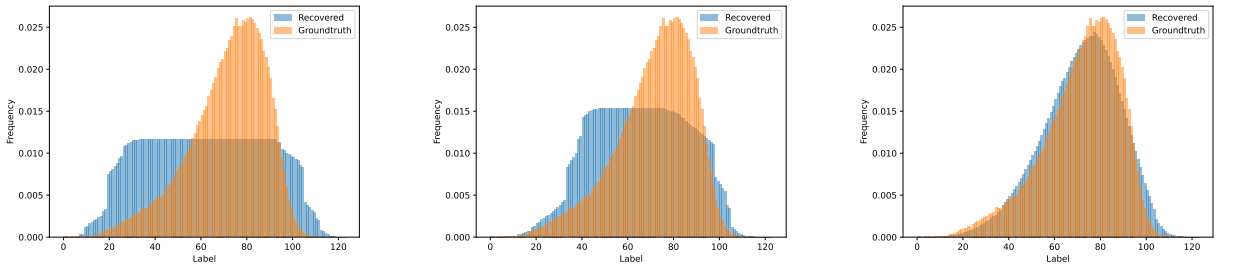
```
1:  procedure AuxQ(rs, b)
2:      ▷ Returns the set of indices of queries that return b.
3:      r ← ∅
4:      for i = 1 to |rs| do
5:          if b ∈ rs[i] then
6:              r ← r ∪ {i}
7:          end if
8:      end for
9:      return r
10: end procedure
11:
12: procedure CreateMinSets(rs, buckets)
13:     minSets, acc ← ∅
14:     prevB ← buckets[1]
15:     for i = 1 to |buckets| do
16:         if AuxQ(rs, buckets[i]) = AuxQ(rs, prevB) then
17:             acc ← acc ∪ {buckets[i]}
18:         else
19:             minSets ← minSets ∪ acc
20:             acc ← ∅
21:         end if
22:         prevB ← buckets[i]
23:     end for
24:     if acc ≠ ∅ then
25:         minSets ← minSets ∪ acc
26:     end if
27:     return minSets
28: end procedure
```

In more detail, we count the level starting from 0. Levels with more buckets (e.g., level 13 as in Figure 4)—and therefore also more records—can contain more minimal sets (for the same amount of queries). This allows for tighter constraints on the interior points of buckets.

For levels that contain only a few buckets, the histogram recovery is inaccurate. Few buckets per level imply that very little information about the bucket's interior point can be learned from queries. This means that each bucket has a far greater possible label range than it would if there were many more records. In Figure 13 in Appendix D.1, we compare the recoveries of level 4, level 9, and level 13 for the 70% queries.

With few records in the dataset, the bucketization technique can better mask the labels as the access leakage from find queries is limited. This is because the total number of buckets is low, thus resulting in less leakage from the access. For the Atlanta Age (Figure 13(b)) and Atlanta Salary (Figure 13(a)) datasets in Appendix D.1, we show this is indeed the case.

Results on Query Recovery. DO-Range does not claim query privacy, arguing that queries for data analytics are typically public. Based on our normalized histogram recovery, we can naturally infer bounds on the query ranges. This can be achieved because the attacker knows which buckets match the queries and the approximate label range of these buckets through the (reconstructed) interior points, allowing it to approximately infer the original query range.



(a) 20% of labels queried, $\overline{\texttt{error}} = 0.6824$ **(b) 40% of labels queried,** $\overline{\texttt{error}} = 0.415$ **(c) 70% of labels queried,** $\overline{\texttt{error}} = 0.1853$

Figure 4: DO-Range**: normalized histogram recovery for the IMDB Age dataset, shown for different percentages of queried labels over $\mathbb{L}$, based on Level 13 of the construction. The error is computed as the average of five runs.**

Utility / Privacy Tradeoff. For the DO-Range scheme to have practical relevance, it is essential that it provides good utility while ensuring meaningful privacy. Unlike conventional DP, DO-Range guarantees perfect correctness, so utility should be measured on bandwidth overhead. This means that the bucket size Z cannot be too large; otherwise, it may fail to accommodate relatively small databases. Moreover, the bucket size Z is the only parameter that can be freely chosen once the label space $\mathbb{L}$ and the upper bound on the number of records N are fixed. Consequently, Z directly impacts the ϵ, δ parameters for differential obliviousness and our DP-Access notion.

In our experiments, we set the bucket size to either 16 or 512. The reason we did not use the bucket size suggested by Chan et al. [11, 12] is that, for $N = 2^{18}$ records (sufficient for our IMDB Age dataset), Z would need to be at least $4.29{\times}10^9$ (see Appendix D for derivation). Such a bucket size would result in a single bucket containing more records than the entire dataset used in our experiments.[11] For a find query that selects a single record, the scheme will always need to transfer at least one bucket. One could argue that this parameter selection is overly conservative to ensure a negligible δ.

To address this, we opted for much more practical bucket sizes of 16 and 512. The bucket size 512, as suggested by [68], guarantees DP access with $\epsilon = 1.0$, $\delta = 10^{-35}$ (using the same notion as in [11, 12]). We demonstrate the recovery results for a bucket size of 512 using synthetic datasets, as shown in Figure 11 in Appendix D.1.

6.3 Construction and Attack on ϵ-OPE-Range

Overview of ϵ-OPE-Range (CCS'22) [16]. Order-preserving encryption (OPE) is a well-studied cryptographic primitive that encrypts messages under a secret key so that the ciphertexts preserve the ordering of the plaintext messages [3]. This order-preserving property enables an untrusted server holding only ciphertexts to support efficient range queries. The strongest security notion of OPE is called IND-FA-OCPA, proposed by Kerschbaum [38]. The term "FA" refers to "frequency-analyzing", meaning that, in addition to the standard order-preserving guarantee, the ciphertexts do not reveal the equality relation. However, leakage-abuse attacks against OPE schemes still persist [27, 46, 50]. The latest development includes Chowdhury et al. [16] putting forward a new notion called ϵ-IND-FA-OCPA, extending IND-FA-OCPA with differential privacy. They propose the ϵ-OPE encryption scheme that satisfies this notion. We refer to the encrypted range query scheme built using ϵ-OPE ciphertexts proposed by them as ϵ-OPE-Range.

Essentially, ϵ-OPE-Range achieves DP access through *partitioning* and *weakening the query correctness*. The major differences between ϵ-OPE-Range and existing range-query schemes built on OPE ciphertexts are:

(1) ϵ-OPE-Range ciphertexts preserve the ordering on the granularity of partitions instead of labels.
(2) ϵ-OPE-Range introduces false positives and false negatives to query results.

Partitions and Encodings. Before describing the construction, we first recall the concept of partitions and encodings [16]. Given a (discrete) label space $\mathbb{L}$, a partition is a set of discrete ranges. As a toy example, considering the discrete range $[1, 100]$, the 4-partition set $\{[1, 25], (25, 50], (50, 75], (75, 100]\}$ is a partition of equal size.

Let us for now use $\mathcal{P}(\cdot)$ to denote the partitioning algorithm. It takes as input a label from $\mathbb{L}$ and outputs the partition index. This partition index is called an "encoding" that lies in some output domain $\mathcal{O}$. More specifically, $\mathcal{P}(\cdot) : \mathbb{L} \to \mathcal{O}$. Continuing with the example of the 4-partition set we mentioned before, let (o_1, o_2, o_3, o_4) be the corresponding encodings of the partition indices. Then this implies $\mathcal{P}(40) = o_2$.

The partitioning algorithm we have discussed so far is deterministic, meaning that there is no noise to guarantee DP yet. Chowdhury et al. propose adding DP to the partitioning algorithm by introducing a randomized encoding mechanism called OPϵc [16]. This randomized OPϵc takes as input $(x, \mathcal{P}, \epsilon)$, where $x \in \mathbb{L}$, some partitioning algorithm $\mathcal{P}$, and privacy budget $\epsilon > 0$, and outputs an encoding in $\mathcal{O}$. For example, OP$\epsilon c(40, \mathcal{P}, \epsilon)$ can output o_2 and possibly o_1 or o_3 as well. The probability distribution of OPϵc depends on ϵ and is selected so that the correct partition index is the most likely output.

Construction of ϵ-OPE-Range. The randomized encoding is then taken as input to a standard OPE scheme to create the ϵ-OPE ciphertext. Further, ϵ-OPE-Range augments the ϵ-OPE ciphertext with encryption of the actual record using standard blockcipher-based

[11] Computed using $\epsilon = 72$, $|\mathbb{L}| = 120$, and a negligible δ. The smaller the ϵ, the greater the Z.

encryption. This allows encrypted records to be compared based on their encrypted random encodings of the labels.

It is clear that the randomized encoding mechanism introduces incorrectness for find queries as the records are inserted into a randomly sampled partition.[12] To mitigate the impact of this and reduce the probability of missing matching records, the client needs to access additional partitions.

We observe that in the extreme case where there is only a single partition, any ordering information is lost, and the scheme's OPE property vanishes. At the other extreme, where each label has its own partition, the scheme approximates traditional OPE; however, it can still misclassify records by assigning them to incorrect partitions.

Range Queries. For range queries, the client first looks up its locally stored partition information to determine the minimal and maximal encodings (i.e., partition indices) of the query range. It then encrypts these two encodings with OPE, used by ϵ-OPE, and sends them to the server. Because of the order-preserving property, the server can compare the ciphertexts and return the encrypted records in the matched partitions. The client decrypts and purges all records that do not fall within the queried range (false positives). To reduce the error from missing records (false negatives),[13] the client can also request the adjacent partitions.

Attack Assumptions. We assume the attacker knows the minimal and maximal labels of the dataset and has access to the encrypted database and the transcripts of find queries. For our experiments, we chose a uniform partition of equal size as suggested in [16], with no prior knowledge of the data. We assume sufficient find queries are made so that each partition is queried at least once.[14] This is necessary if the underlying OPE satisfies IND-FA-OCPA. For the weaker OPE notion that reveals the equality relation, no find queries are needed.

Attack Description. At a high level, this attack recovers the partition boundaries using the CreateMinSets procedure described in our DO-Range attack. From there, we can straightforwardly compute the histogram. The main difference compared with the DO-Range attack is that there are no longer dummy records. However, the recovery can be inaccurate due to insertion noise.

With this attack, we can recover the approximate assignment of records to partitions and the rough ordering of records at the granularity of the partitions. It is important to note that we cannot recover the exact assignment, as each record has a small probability of being inserted into incorrect partitions (likely adjacent ones). However, ϵ-OPE-Range must keep this error probability small to achieve high query correctness.

We present the pseudocode for the attack in Algorithm 3. The main Attack procedure takes as input a list of query responses `rs`, and the ciphertexts `ct` stored on the server.

The ciphertexts `ct` are first sorted (Line 7) and the minimal sets are recovered (Line 8). The label space $\mathbb{L}$ is divided into equal-sized parts of length l (Line 12). Next, the AssignedLabels procedure iterates over the partitions (as recovered by CreateMinSets) and assigns the next l labels to it (Line 15).

The attack complexity is $O(\max\{\log N, m \log m\} \cdot N)$ for N records and m partitions, considering the sorting procedure and the partition recovery using the CreateMinSets procedure.

Algorithm 3 ϵ-OPE-Range Attack

```
1:  procedure Attack(rs, ct)
2:      parts ← RecoverPartitions(rs, ct)
3:      labels ← AssignedLabels(parts, 𝕃)
4:  end procedure
5:
6:  procedure RecoverPartitions(rs, ct)
7:      ct ← OPE.Sort(ct)
8:      parts ← CreateMinSets(rs, ct)
9:  end procedure
10:
11: procedure AssignedLabels(parts, 𝕃)
12:     labels ← {}; cnt ← 1; l ← |𝕃| / |parts|
13:     for i = 1 to |parts| do
14:         ▷ Assign the next l labels to partition i
15:         labels[i] ← 𝕃[cnt : cnt + l]
16:         cnt ← cnt + l
17:     end for
18:     return labels
19: end procedure
```

Results on Data Recovery. We show the histogram recovery for the IMDB Age dataset in Figure 5, and that of Atlanta Age and Salary datasets in Figure 14 in Appendix D.2. We compute the `error` term as defined in Section 5. The accuracy of the recovery depends on the number of partitions and privacy budget ϵ, which determines the insertion noise, the probability that a record is inserted into an "incorrect" partition.[15]

First, let us look at the privacy budget ϵ for DP access. For a fixed partition of equal size, as we expected, a smaller ϵ provides a better privacy guarantee. This is observed in Figure 5(b), where with $\epsilon = 0.5$ the histogram recovery is less accurate compared with Figure 5(c) with $\epsilon = 5$. We provide more results for varying ϵ in Figures 14(d), 14(e), and 14(f) in Appendix D.2.

As for varying the number of partitions for fixed ϵ, we found that more partitions means that the ordering information is preserved at a finer granularity and, therefore, the histogram is more accurate. We show this correlation in Figures 14(a), 14(b) and 14(c) in Appendix D.2.

Similar to the "bucket" concept in DO-Range, each partition in ϵ-OPE-Range is the smallest data unit accessed by the queries. Therefore, information-theoretically, it is not possible to learn the exact values of individual records within the partition.

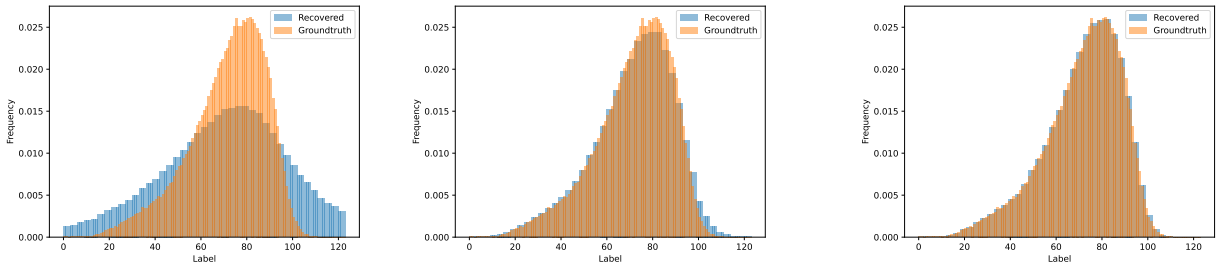
Our experimental results also reveal an interesting finding: the small insertion noise added by ϵ-OPE-Range for DP access often cancels out and thus does not significantly affect our histogram recovery. Specifically, assume there are two records that should be

[12] The probability distribution depends on ϵ.

[13] False negatives refer to matching records that were not returned since the encoding assigned them to partitions outside the client's requested range.

[14] The expected number of queries needed is $m \log m$ for m partitions with uniformly random queries.

[15] Although δ appears in the security definition ϵ-IND-FA-OCPA, but is not used in ϵ-OPE-Range [16].



(a) $\epsilon = 0.1$, $\overline{\texttt{error}} = 0.4861$ **(b)** $\epsilon = 0.5$, $\overline{\texttt{error}} = 0.0798$ **(c)** $\epsilon = 5$, $\overline{\texttt{error}} = 0.0396$

Figure 5: Normalized histogram recovery for ϵ-OPE-Range on the IMDB Age dataset: 41 partitions, with 70% of labels queried; retrieving two neighbouring partitions. The error is computed as the average of five runs.

assigned adjacent partitions. If both are instead assigned to each other's partitions, the histogram remains unchanged. Over time, this insertion noise smooths out, and our recovery takes advantage of this fact. We note that in practice, this insertion noise also needs to be small to have good utility. We discuss this aspect further in the utility/privacy tradeoff.

Results on Query Recovery. The attacker learns which partitions are retrieved and therefore can directly infer the find query range at the granularity of the partition bounds that we can recover.

Utility/Privacy Tradeoff. In ϵ-OPE-Range, the number of partitions and ϵ must be chosen in such a way that most records fall into their correct partitions, otherwise, either the efficiency or the query correctness would suffer significantly. Further, if the scheme is restricted to only use a small number of partitions, then clients will have to retrieve a large fraction of the database for every find query. Therefore, an efficient instantiation should have a small partition size, meaning a larger number of partitions, to keep the bandwidth overhead low.

In order to keep the number of missing records for queries low, additional partitions on the left and right side of the "matching" partition can be retrieved, which increases the bandwidth overhead. We show this number of missing records for point queries on the IMDB age dataset for various number of additionally included partitions in Figure 12.

6.4 The Gap

In this section, we discuss the gap and its implications between differential privacy as applied in data analytics and when it is used in private data access (DP-Access).

Privacy Budget and Failure Probability. The privacy budget, represented by ϵ, restricts the number of queries in DP analytics. Intuitively, as the number of queries increases, so does the privacy loss. However, our studies on DO-Range and ϵ-OPE-Range yield a surprising result: the privacy loss defined in DP access comes only from the insertions. Find queries are not bounded by the privacy budget ϵ at all. As for the failure probability δ, it should be *negligible* in the database size to ensure individual privacy. This greatly affects the utility aspect of DO-Range, resulting in a non-practical bucket size. In comparison, ϵ-OPE-Range does not have a failure probability δ, so there is no impact.[16]

Utility and Efficiency/Correctness. In DP analytics, high utility implies that the noise added for privacy does not significantly impact the accuracy of data analytics. Given the absence of a formal definition, utility is often measured on a case-by-case basis. In contrast, when it comes to DP access, efficiency emerges as the primary factor influencing utility. We illustrate this point using our case studies on the range-query data structures DO-Range and ϵ-OPE-Range.

For DO-Range, since the correctness of range queries is ensured, utility is closely linked to bandwidth cost, particularly the bucket size. As mentioned in the utility/privacy tradeoff (Section 6.2), a bucket size greater than the entire dataset certainly does not achieve good utility. For ϵ-OPE-Range, utility should be measured based on both the bandwidth cost and the accuracy of the results, as perfect correctness is not guaranteed. A small partition number indicates that a large fraction of the database is returned. The implication for correctness is also substantial: in ϵ-OPE-Range, certain records cannot even be retrieved through find queries.

Techniques. A notable distinction between DP analytics and DP access lies in the techniques used. Specifically, for DP analytics, on each query, the noise added through the Laplace mechanism (cf. Section 2.1) is calibrated based on the sensitivity of the function. But for functions such as range queries, sensitivity is theoretically unbounded, or at least the size of the entire database, making it infeasible. In DP access, techniques like bucketization and partitioning are used instead.

DO-Range uses a bucketization approach in its construction, while ϵ-OPE-Range relies on partitioning. In Table 2, we compare these two techniques. Notably, DO-Range's DP access guarantee comes from the fact that the real number of records is unknown to the adversary due to the padding with dummy records. In contrast, ϵ-OPE-Range 's DP access guarantee comes instead from two sources: hiding the exact number of records in a partition as the insertions are probabilistic, and allowing some failure in retrieving all matching records.

[16]The security definition ϵ-IND-FA-OCPA however includes δ.

	Bucketization	Partitioning
Fixed bucket/partition size	✓	✗
Flexible number of real records	✓	✓
Adding dummy records	✓	✗
Privacy by hiding the number of real records	✓	✓
Privacy by sacrificing query correctness	✗	✓

Table 2: Comparison of bucketization and partitioning.

Query and Data Privacy. Recall from Section 3 that in a wide range of applications related to private databases, both query and data privacy are essential. These two privacy requirements align with STE's security definition. In response to leakage-abuse attacks, there has been a trend towards adopting DP access as a strategy to mitigate access leakage. This is also evident in our case studies; both DO-Range and ϵ-OPE-Range encrypt the data with a semantically-secure encryption scheme and use DP access. We demonstrate that either the efficiency/correctness cost is significant, or that the notion itself is ineffective in guaranteeing strong query and data privacy.

7 Related Work

The literature on encrypted search (ESAs) is extensive. Here, we highlight the most representative work in searchable symmetric encryption (SSE) [5, 6, 9, 10, 18, 19, 53, 63, 64] and its generalization, structured encryption (STE) [13, 17]. For a more comprehensive overview, we direct readers to survey results [24].

Regarding the leakage-abuse attack techniques used in our studies, there are quite a few related work, including those exploiting volume, access, and search patterns [8, 26, 32, 50, 57, 67]. We focus on those related to range queries in our analysis [26, 28, 40, 42, 47, 48, 56]. A survey on the effectiveness of related attacks can be found in [33].

Similarly, extensive literature can be found on ORAM [1, 21, 61, 65] and differential privacy (DP) [20, 46]. Of particular relevance to our work are: differentially-private ORAM [66], oblivious data structures with DP such as [2, 60]; lower bound results on DP ORAM, and other data structures [54, 55].

Another line of research applies DP to ESAs with equality searches (i.e., point queries) [14, 52, 62]. This is followed by cryptanalysis studies exploiting additional search patterns [51] or auxiliary distribution information [29].

The bucketization technique, which we study in this work, has also been adopted for leakage mitigation in [34] and [30, 39]. Finally, recent studies on quantifying leakage can be found in [4, 35, 39].

8 Conclusion

In this work, we study the secure access mechanism to a private database with differential privacy. Research on database privacy is not new — there is extensive work on encrypted databases with structured encryption (STE) and searchable symmetric encryption (SSE), as well as differential privacy (DP) in data analytics. However, the application of DP in accessing a private database is a more recent exploration.

For our analysis, we first categorize various applications related to private databases based on their privacy guarantees. We then define the suitable threat models and adversarial goals, capturing real-world attacks. To analyze DP access, we formalize adaptive and non-adaptive DP-Access notions that are compatible with the standard STE syntax. These notions are general to fit any encrypted database or data structure. They capture the schemes we study, DO-Range and ϵ-OPE-Range. We have identified two generic techniques, *bucketization* and *partitioning* to achieve DP access. Finally, we address the gap between DP analytics and DP access in terms of efficiency, correctness, and privacy.

Our findings indicate that DP access is still far from being a secure access mechanism that offers strong privacy for private data. It does not come at a low cost, often compromising efficiency or correctness. The DP access leakage still reveals significant information about the queries and data. Nonetheless, rigorously studying access leakage should be the route to a more secure accessing mechanism; DP access represents only the initial step.

Ethics Considerations

This work complies with the ethics guideline. It aligns with the intended use of the analyzed constructions: DO-Range and ϵ-OPE-Range, and it does not expose any vulnerabilities in the constructions or underlying cryptographic primitives. To the best of our knowledge, there is no real-world deployment of these constructions.

Use of Public and Synthetic Data. Our experiments were conducted using publicly available IMDB and Atlanta datasets and synthetic datasets. We ensured that no private information was utilized or exposed, thereby eliminating any risks of harm to individuals or groups.

Artifact Availability

As attested in the submission form, we are currently participating in the artifact evaluation process. We will make the artifact for the experimental evaluation available at https://doi.org/10.5281/zenodo.17035133.

Acknowledgements

The authors are grateful to the ACM CCS '25 reviewers for their helpful comments. Tianxin Tang is supported by an NWO VIDI grant (Project No. VI.Vidi.193.066).

References

[1] Gilad Asharov, Ilan Komargodski, Wei-Kai Lin, Kartik Nayak, Enoch Peserico, and Elaine Shi. 2020. OptORAMa: Optimal Oblivious RAM. In *Advances in Cryptology – EUROCRYPT 2020, Part II (Lecture Notes in Computer Science, Vol. 12106)*, Anne Canteaut and Yuval Ishai (Eds.). Springer, Cham, Switzerland, Zagreb, Croatia, 403–432. doi:10.1007/978-3-030-45724-2_14

[2] Dmytro Bogatov, Georgios Kellaris, George Kollios, Kobbi Nissim, and Adam O'Neill. 2021. ϵpsolute: Efficiently Querying Databases While Providing Differential Privacy. In *ACM CCS 2021: 28th Conference on Computer and Communications Security*, Giovanni Vigna and Elaine Shi (Eds.). ACM Press, Virtual Event, Republic of Korea, 2262–2276. doi:10.1145/3460120.3484786

[3] Alexandra Boldyreva, Nathan Chenette, Younho Lee, and Adam O'Neill. 2009. Order-Preserving Symmetric Encryption. In *Advances in Cryptology – EUROCRYPT 2009 (Lecture Notes in Computer Science, Vol. 5479)*, Antoine Joux (Ed.). Springer Berlin Heidelberg, Germany, Cologne, Germany, 224–241. doi:10.1007/978-3-642-01001-9_13

[4] Alexandra Boldyreva, Zichen Gui, and Bogdan Warinschi. 2024. Understanding Leakage in Searchable Encryption: a Quantitative Approach. *Proceedings on Privacy Enhancing Technologies* 2024, 4 (Oct. 2024), 503–524. doi:10.56553/popets-2024-0127

[5] Raphael Bost. 2016. $\Sigma o \phi o \varsigma$: Forward Secure Searchable Encryption. In *ACM CCS 2016: 23rd Conference on Computer and Communications Security*, Edgar R. Weippl,

Stefan Katzenbeisser, Christopher Kruegel, Andrew C. Myers, and Shai Halevi (Eds.). ACM Press, Vienna, Austria, 1143–1154. doi:10.1145/2976749.2978303

[6] Raphaël Bost, Brice Minaud, and Olga Ohrimenko. 2017. Forward and Backward Private Searchable Encryption from Constrained Cryptographic Primitives. In *ACM CCS 2017: 24th Conference on Computer and Communications Security*, Bhavani M. Thuraisingham, David Evans, Tal Malkin, and Dongyan Xu (Eds.). ACM Press, Dallas, TX, USA, 1465–1482. doi:10.1145/3133956.3133980

[7] Mark Bun, Kobbi Nissim, Uri Stemmer, and Salil P. Vadhan. 2015. Differentially Private Release and Learning of Threshold Functions. In *56th Annual Symposium on Foundations of Computer Science*, Venkatesan Guruswami (Ed.). IEEE Computer Society Press, Berkeley, CA, USA, 634–649. doi:10.1109/FOCS.2015.45

[8] David Cash, Paul Grubbs, Jason Perry, and Thomas Ristenpart. 2015. Leakage-Abuse Attacks Against Searchable Encryption. In *ACM CCS 2015: 22nd Conference on Computer and Communications Security*, Indrajit Ray, Ninghui Li, and Christopher Kruegel (Eds.). ACM Press, Denver, CO, USA, 668–679. doi:10.1145/2810103.2813700

[9] David Cash, Joseph Jaeger, Stanislaw Jarecki, Charanjit S. Jutla, Hugo Krawczyk, Marcel-Catalin Rosu, and Michael Steiner. 2014. Dynamic Searchable Encryption in Very-Large Databases: Data Structures and Implementation. In *ISOC Network and Distributed System Security Symposium – NDSS 2014*. The Internet Society, San Diego, CA, USA. doi:10.14722/ndss.2014.23264

[10] Javad Ghareh Chamani, Dimitrios Papadopoulos, Charalampos Papamanthou, and Rasool Jalili. 2018. New Constructions for Forward and Backward Private Symmetric Searchable Encryption. In *ACM CCS 2018: 25th Conference on Computer and Communications Security*, David Lie, Mohammad Mannan, Michael Backes, and XiaoFeng Wang (Eds.). ACM Press, Toronto, ON, Canada, 1038–1055. doi:10.1145/3243734.3243833

[11] T.-H. Hubert Chan, Kai-Min Chung, Bruce Maggs, and Elaine Shi. 2022. Foundations of Differentially Oblivious Algorithms. *J. ACM* 69, 4, Article 27 (Aug. 2022), 49 pages. doi:10.1145/3555984

[12] T.-H. Hubert Chan, Kai-Min Chung, Bruce M. Maggs, and Elaine Shi. 2019. Foundations of Differentially Oblivious Algorithms. In *30th Annual ACM-SIAM Symposium on Discrete Algorithms*, Timothy M. Chan (Ed.). ACM-SIAM, San Diego, CA, USA, 2448–2467. doi:10.1137/1.9781611975482.150

[13] Melissa Chase and Seny Kamara. 2010. Structured Encryption and Controlled Disclosure. In *Advances in Cryptology – ASIACRYPT 2010 (Lecture Notes in Computer Science, Vol. 6477)*, Masayuki Abe (Ed.). Springer Berlin Heidelberg, Germany, Singapore, 577–594. doi:10.1007/978-3-642-17373-8_33

[14] Guoxing Chen, Ten-Hwang Lai, Michael K. Reiter, and Yinqian Zhang. 2018. Differentially Private Access Patterns for Searchable Symmetric Encryption. In *IEEE INFOCOM 2018 - IEEE Conference on Computer Communications*. IEEE, Honolulu, HI, 810–818. doi:10.1109/INFOCOM.2018.8486381

[15] Junjie Chen, Wendy Hui Wang, and Xinghua Shi. 2020. Differential Privacy Protection Against Membership Inference Attack on Machine Learning for Genomic Data. In *Biocomputing 2021*. WORLD SCIENTIFIC, 26–37. doi:10.1142/9789811232701_0003

[16] Amrita Roy Chowdhury, Bolin Ding, Somesh Jha, Weiran Liu, and Jingren Zhou. 2022. Strengthening Order Preserving Encryption with Differential Privacy. In *ACM CCS 2022: 29th Conference on Computer and Communications Security*, Heng Yin, Angelos Stavrou, Cas Cremers, and Elaine Shi (Eds.). ACM Press, Los Angeles, CA, USA, 2519–2533. doi:10.1145/3548606.3560610

[17] Reza Curtmola, Juan A. Garay, Seny Kamara, and Rafail Ostrovsky. 2006. Searchable symmetric encryption: improved definitions and efficient constructions. In *ACM CCS 2006: 13th Conference on Computer and Communications Security*, Ari Juels, Rebecca N. Wright, and Sabrina De Capitani di Vimercati (Eds.). ACM Press, Alexandria, Virginia, USA, 79–88. doi:10.1145/1180405.1180417

[18] Ioannis Demertzis, Javad Ghareh Chamani, Dimitrios Papadopoulos, and Charalampos Papamanthou. 2020. Dynamic Searchable Encryption with Small Client Storage. In *ISOC Network and Distributed System Security Symposium – NDSS 2020*. The Internet Society, San Diego, CA, USA. doi:10.14722/ndss.2020.24423

[19] Ioannis Demertzis, Dimitrios Papadopoulos, Charalampos Papamanthou, and Saurabh Shintre. 2020. SEAL: Attack Mitigation for Encrypted Databases via Adjustable Leakage. In *USENIX Security 2020: 29th USENIX Security Symposium*, Srdjan Capkun and Franziska Roesner (Eds.). USENIX Association, 2433–2450.

[20] Damien Desfontaines and Balázs Pejó. 2020. SoK: Differential privacies. *Proceedings on Privacy Enhancing Technologies* 2020, 2 (April 2020), 288–313. doi:10.2478/popets-2020-0028

[21] Srinivas Devadas, Marten van Dijk, Christopher W. Fletcher, Ling Ren, Elaine Shi, and Daniel Wichs. 2016. Onion ORAM: A Constant Bandwidth Blowup Oblivious RAM. In *TCC 2016-A: 13th Theory of Cryptography Conference, Part II (Lecture Notes in Computer Science, Vol. 9563)*, Eyal Kushilevitz and Tal Malkin (Eds.). Springer Berlin Heidelberg, Germany, Tel Aviv, Israel, 145–174. doi:10.1007/978-3-662-49099-0_6

[22] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. 2006. Calibrating Noise to Sensitivity in Private Data Analysis. In *TCC 2006: 3rd Theory of Cryptography Conference (Lecture Notes in Computer Science, Vol. 3876)*, Shai Halevi and Tal Rabin (Eds.). Springer Berlin Heidelberg, Germany, New York, NY, USA, 265–284. doi:10.1007/11681878_14

[23] Cynthia Dwork and Aaron Roth. 2013. The Algorithmic Foundations of Differential Privacy. *Foundations and Trends® in Theoretical Computer Science* 9, 3-4 (2013), 211–407. doi:10.1561/0400000042

[24] Benjamin Fuller, Mayank Varia, Arkady Yerukhimovich, Emily Shen, Ariel Hamlin, Vijay Gadepally, Richard Shay, John Darby Mitchell, and Robert K. Cunningham. 2017. SoK: Cryptographically Protected Database Search. In *2017 IEEE Symposium on Security and Privacy*. IEEE Computer Society Press, San Jose, CA, USA, 172–191. doi:10.1109/SP.2017.10

[25] Marilyn George, Seny Kamara, and Tarik Moataz. 2021. Structured Encryption and Dynamic Leakage Suppression. In *Advances in Cryptology – EUROCRYPT 2021, Part III (Lecture Notes in Computer Science, Vol. 12698)*, Anne Canteaut and François-Xavier Standaert (Eds.). Springer, Cham, Switzerland, Zagreb, Croatia, 370–396. doi:10.1007/978-3-030-77883-5_13

[26] Paul Grubbs, Marie-Sarah Lacharité, Brice Minaud, and Kenneth G. Paterson. 2018. Pump up the Volume: Practical Database Reconstruction from Volume Leakage on Range Queries. In *ACM CCS 2018: 25th Conference on Computer and Communications Security*, David Lie, Mohammad Mannan, Michael Backes, and XiaoFeng Wang (Eds.). ACM Press, Toronto, ON, Canada, 315–331. doi:10.1145/3243734.3243864

[27] Paul Grubbs, Kevin Sekniqi, Vincent Bindschaedler, Muhammad Naveed, and Thomas Ristenpart. 2017. Leakage-Abuse Attacks against Order-Revealing Encryption. In *2017 IEEE Symposium on Security and Privacy*. IEEE Computer Society Press, San Jose, CA, USA, 655–672. doi:10.1109/SP.2017.44

[28] Zichen Gui, Oliver Johnson, and Bogdan Warinschi. 2019. Encrypted Databases: New Volume Attacks against Range Queries. In *ACM CCS 2019: 26th Conference on Computer and Communications Security*, Lorenzo Cavallaro, Johannes Kinder, XiaoFeng Wang, and Jonathan Katz (Eds.). ACM Press, London, UK, 361–378. doi:10.1145/3319535.3363210

[29] Zichen Gui, Kenneth G. Paterson, and Sikhar Patranabis. 2023. Rethinking Searchable Symmetric Encryption. In *2023 IEEE Symposium on Security and Privacy*. IEEE Computer Society Press, San Francisco, CA, USA, 1401–1418. doi:10.1109/SP46215.2023.10179460

[30] Zichen Gui, Kenneth G. Paterson, Sikhar Patranabis, and Bogdan Warinschi. 2024. SWiSSSE: System-Wide Security for Searchable Symmetric Encryption. *Proceedings on Privacy Enhancing Technologies* 2024, 1 (Jan. 2024), 549–581. doi:10.56553/popets-2024-0032

[31] Trung Ha, Trang Vo, Tran Khanh Dang, and Nguyen Thi Huyen Trang. 2023. Differential Privacy under Membership Inference Attacks. In *Future Data and Security Engineering. Big Data, Security and Privacy, Smart City and Industry 4.0 Applications*, Tran Khanh Dang, Josef Küng, and Tai M. Chung (Eds.). Springer Nature Singapore, Singapore, 255–269.

[32] Mohammad Saiful Islam, Mehmet Kuzu, and Murat Kantarcioglu. 2012. Access Pattern disclosure on Searchable Encryption: Ramification, Attack and Mitigation. In *ISOC Network and Distributed System Security Symposium – NDSS 2012*. The Internet Society, San Diego, CA, USA.

[33] Seny Kamara, Abdelkarim Kati, Tarik Moataz, Thomas Schneider, Amos Treiber, and Michael Yonli. 2022. SoK: Cryptanalysis of Encrypted Search with LEAKER - A framework for LEakage AttacK Evaluation on Real-world data. In *2022 IEEE European Symposium on Security and Privacy*. IEEE Computer Society Press, Genoa, Italy, 90–108. doi:10.1109/EuroSP53844.2022.00014

[34] Seny Kamara and Tarik Moataz. 2019. Computationally Volume-Hiding Structured Encryption. In *Advances in Cryptology – EUROCRYPT 2019, Part II (Lecture Notes in Computer Science, Vol. 11477)*, Yuval Ishai and Vincent Rijmen (Eds.). Springer, Cham, Switzerland, Darmstadt, Germany, 183–213. doi:10.1007/978-3-030-17656-3_7

[35] Seny Kamara and Tarik Moataz. 2023. Bayesian Leakage Analysis: A Framework for Analyzing Leakage in Encrypted Search. Cryptology ePrint Archive, Report 2023/813. https://eprint.iacr.org/2023/813

[36] Seny Kamara, Tarik Moataz, and Olga Ohrimenko. 2018. Structured Encryption and Leakage Suppression. In *Advances in Cryptology – CRYPTO 2018, Part I (Lecture Notes in Computer Science, Vol. 10991)*, Hovav Shacham and Alexandra Boldyreva (Eds.). Springer, Cham, Switzerland, Santa Barbara, CA, USA, 339–370. doi:10.1007/978-3-319-96884-1_12

[37] Georgios Kellaris, George Kollios, Kobbi Nissim, and Adam O'Neill. 2016. Generic Attacks on Secure Outsourced Databases. In *ACM CCS 2016: 23rd Conference on Computer and Communications Security*, Edgar R. Weippl, Stefan Katzenbeisser, Christopher Kruegel, Andrew C. Myers, and Shai Halevi (Eds.). ACM Press, Vienna, Austria, 1329–1340. doi:10.1145/2976749.2978386

[38] Florian Kerschbaum. 2015. Frequency-Hiding Order-Preserving Encryption. In *ACM CCS 2015: 22nd Conference on Computer and Communications Security*, Indrajit Ray, Ninghui Li, and Christopher Kruegel (Eds.). ACM Press, Denver, CO, USA, 656–667. doi:10.1145/2810103.2813629

[39] Evgenios M. Kornaropoulos, Nathaniel Moyer, Charalampos Papamanthou, and Alexandros Psomas. 2022. Leakage Inversion: Towards Quantifying Privacy in Searchable Encryption. In *ACM CCS 2022: 29th Conference on Computer and Communications Security*, Heng Yin, Angelos Stavrou, Cas Cremers, and Elaine Shi (Eds.). ACM Press, Los Angeles, CA, USA, 1829–1842. doi:10.1145/3548606.3560593

[40] Evgenios M. Kornaropoulos, Charalampos Papamanthou, and Roberto Tamassia. 2020. The State of the Uniform: Attacks on Encrypted Databases Beyond the Uniform Query Distribution. In *2020 IEEE Symposium on Security and Privacy*. IEEE Computer Society Press, San Francisco, CA, USA, 1223–1240. doi:10.1109/SP40000.2020.00029

[41] Evgenios M. Kornaropoulos, Charalampos Papamanthou, and Roberto Tamassia. 2021. Response-Hiding Encrypted Ranges: Revisiting Security via Parametrized Leakage-Abuse Attacks. In *2021 IEEE Symposium on Security and Privacy*. IEEE Computer Society Press, San Francisco, CA, USA, 1502–1519. doi:10.1109/SP40001.2021.00044

[42] Marie-Sarah Lacharité, Brice Minaud, and Kenneth G. Paterson. 2018. Improved Reconstruction Attacks on Encrypted Data Using Range Query Leakage. In *2018 IEEE Symposium on Security and Privacy*. IEEE Computer Society Press, San Francisco, CA, USA, 297–314. doi:10.1109/SP.2018.00002

[43] Marie-Sarah Lacharite, Brice Minaud, and Kenneth G. Paterson. 2018. Improved Reconstruction Attacks on Encrypted Data Using Range Query Leakage. In *2018 IEEE Symposium on Security and Privacy (SP)*. IEEE, San Francisco, CA, 297–314. doi:10.1109/SP.2018.00002

[44] Kasper Green Larsen and Jesper Buus Nielsen. 2018. Yes, There is an Oblivious RAM Lower Bound!. In *Advances in Cryptology – CRYPTO 2018, Part II (Lecture Notes in Computer Science, Vol. 10992)*, Hovav Shacham and Alexandra Boldyreva (Eds.). Springer, Cham, Switzerland, Santa Barbara, CA, USA, 523–542. doi:10.1007/978-3-319-96881-0_18

[45] Johan Lokna, Anouk Paradis, Dimitar I. Dimitrov, and Martin T. Vechev. 2023. Group and Attack: Auditing Differential Privacy. In *ACM CCS 2023: 30th Conference on Computer and Communications Security*, Weizhi Meng, Christian Damsgaard Jensen, Cas Cremers, and Engin Kirda (Eds.). ACM Press, Copenhagen, Denmark, 1905–1918. doi:10.1145/3576915.3616607

[46] Matteo Maffei, Manuel Reinert, and Dominique Schröder. 2017. On the Security of Frequency-Hiding Order-Preserving Encryption. In *CANS 17: 16th International Conference on Cryptology and Network Security (Lecture Notes in Computer Science, Vol. 11261)*, Srdjan Capkun and Sherman S. M. Chow (Eds.). Springer, Cham, Switzerland, Hong Kong, China, 51–70. doi:10.1007/978-3-030-02641-7_3

[47] Evangelia Anna Markatou, Francesca Falzon, Roberto Tamassia, and William Schor. 2021. Reconstructing with Less: Leakage Abuse Attacks in Two Dimensions. In *ACM CCS 2021: 28th Conference on Computer and Communications Security*, Giovanni Vigna and Elaine Shi (Eds.). ACM Press, Virtual Event, Republic of Korea, 2243–2261. doi:10.1145/3460120.3484552

[48] Evangelia Anna Markatou and Roberto Tamassia. 2024. Reconstructing with Even Less: Amplifying Leakage and Drawing Graphs. In *ACM CCS 2024: 31st Conference on Computer and Communications Security*, Bo Luo, Xiaojing Liao, Jun Xu, Engin Kirda, and David Lie (Eds.). ACM Press, Salt Lake City, UT, USA, 4777–4791. doi:10.1145/3658644.3670313

[49] Pratyush Mishra, Rishabh Poddar, Jerry Chen, Alessandro Chiesa, and Raluca Ada Popa. 2018. Oblix: An Efficient Oblivious Search Index. In *2018 IEEE Symposium on Security and Privacy*. IEEE Computer Society Press, San Francisco, CA, USA, 279–296. doi:10.1109/SP.2018.00045

[50] Muhammad Naveed, Seny Kamara, and Charles V. Wright. 2015. Inference Attacks on Property-Preserving Encrypted Databases. In *ACM CCS 2015: 22nd Conference on Computer and Communications Security*, Indrajit Ray, Ninghui Li, and Christopher Kruegel (Eds.). ACM Press, Denver, CO, USA, 644–655. doi:10.1145/2810103.2813651

[51] Simon Oya and Florian Kerschbaum. 2021. Hiding the Access Pattern is Not Enough: Exploiting Search Pattern Leakage in Searchable Encryption. In *USENIX Security 2021: 30th USENIX Security Symposium*, Michael Bailey and Rachel Greenstadt (Eds.). USENIX Association, 127–142.

[52] Sarvar Patel, Giuseppe Persiano, Kevin Yeo, and Moti Yung. 2019. Mitigating Leakage in Secure Cloud-Hosted Data Structures: Volume-Hiding for Multi-Maps via Hashing. In *ACM CCS 2019: 26th Conference on Computer and Communications Security*, Lorenzo Cavallaro, Johannes Kinder, XiaoFeng Wang, and Jonathan Katz (Eds.). ACM Press, London, UK, 79–93. doi:10.1145/3319535.3354213

[53] Sikhar Patranabis and Debdeep Mukhopadhyay. 2021. Forward and Backward Private Conjunctive Searchable Symmetric Encryption. In *ISOC Network and Distributed System Security Symposium – NDSS 2021*. The Internet Society, Virtual. doi:10.14722/ndss.2021.23116

[54] Giuseppe Persiano and Kevin Yeo. 2019. Lower Bounds for Differentially Private RAMs. In *Advances in Cryptology – EUROCRYPT 2019, Part I (Lecture Notes in Computer Science, Vol. 11476)*, Yuval Ishai and Vincent Rijmen (Eds.). Springer, Cham, Switzerland, Darmstadt, Germany, 404–434. doi:10.1007/978-3-030-17653-2_14

[55] Giuseppe Persiano and Kevin Yeo. 2023. Lower Bound Framework for Differentially Private and Oblivious Data Structures. In *Advances in Cryptology – EUROCRYPT 2023, Part I (Lecture Notes in Computer Science, Vol. 14004)*, Carmit Hazay and Martijn Stam (Eds.). Springer, Cham, Switzerland, Lyon, France, 487–517. doi:10.1007/978-3-031-30545-0_17

[56] Rishabh Poddar, Stephanie Wang, Jianan Lu, and Raluca Ada Popa. 2020. Practical Volume-Based Attacks on Encrypted Databases. In *2020 IEEE European Symposium on Security and Privacy*. IEEE Computer Society Press, Genoa, Italy, 354–369. doi:10.1109/EuroSP48549.2020.00030

[57] David Pouliot and Charles V. Wright. 2016. The Shadow Nemesis: Inference Attacks on Efficiently Deployable, Efficiently Searchable Encryption. In *ACM CCS 2016: 23rd Conference on Computer and Communications Security*, Edgar R. Weippl, Stefan Katzenbeisser, Christopher Kruegel, Andrew C. Myers, and Shai Halevi (Eds.). ACM Press, Vienna, Austria, 1341–1352. doi:10.1145/2976749.2978401

[58] Lianke Qin, Rajesh Jayaram, Elaine Shi, Zhao Song, Danyang Zhuo, and Shumo Chu. 2022. Adore: Differentially Oblivious Relational Database Operators. *Proc. VLDB Endow.* 16, 4, 842–855. doi:10.14778/3574245.3574267

[59] Lina Qiu, Georgios Kellaris, Nikos Mamoulis, Kobbi Nissim, and George Kollios. 2023. Doquet: Differentially Oblivious Range and Join Queries with Private Data Structures. *Proc. VLDB Endow.* 16, 13, 4160–4173. doi:10.14778/3625054.3625055

[60] Leonie Reichert, Gowri R. Chandran, Phillipp Schoppmann, Thomas Schneider, and Björn Scheuermann. 2024. Menhir: An Oblivious Database with Protection against Access and Volume Pattern Leakage. In *ASIACCS 24: 19th ACM Symposium on Information, Computer and Communications Security*, Jianying Zhou, Tony Q. S. Quek, Debin Gao, and Alvaro A. Cárdenas (Eds.). ACM Press, Singapore. doi:10.1145/3634737.3657005

[61] Ling Ren, Christopher W. Fletcher, Albert Kwon, Emil Stefanov, Elaine Shi, Marten van Dijk, and Srinivas Devadas. 2015. Constants Count: Practical Improvements to Oblivious RAM. In *USENIX Security 2015: 24th USENIX Security Symposium*, Jaeyeon Jung and Thorsten Holz (Eds.). USENIX Association, Washington, DC, USA, 415–430.

[62] Zhiwei Shang, Simon Oya, Andreas Peter, and Florian Kerschbaum. 2021. Obfuscated Access and Search Patterns in Searchable Encryption. In *ISOC Network and Distributed System Security Symposium – NDSS 2021*. The Internet Society, Virtual. doi:10.14722/ndss.2021.23041

[63] Dawn Xiaodong Song, David Wagner, and Adrian Perrig. 2000. Practical Techniques for Searches on Encrypted Data. In *2000 IEEE Symposium on Security and Privacy*. IEEE Computer Society Press, Oakland, CA, USA, 44–55. doi:10.1109/SECPRI.2000.848445

[64] Emil Stefanov, Charalampos Papamanthou, and Elaine Shi. 2014. Practical Dynamic Searchable Encryption with Small Leakage. In *ISOC Network and Distributed System Security Symposium – NDSS 2014*. The Internet Society, San Diego, CA, USA. doi:10.14722/ndss.2014.23298

[65] Emil Stefanov, Marten van Dijk, Elaine Shi, Christopher W. Fletcher, Ling Ren, Xiangyao Yu, and Srinivas Devadas. 2013. Path ORAM: an extremely simple oblivious RAM protocol. In *ACM CCS 2013: 20th Conference on Computer and Communications Security*, Ahmad-Reza Sadeghi, Virgil D. Gligor, and Moti Yung (Eds.). ACM Press, Berlin, Germany, 299–310. doi:10.1145/2508859.2516660

[66] Sameer Wagh, Paul Cuff, and Prateek Mittal. 2018. Differentially Private Oblivious RAM. *Proceedings on Privacy Enhancing Technologies* 2018, 4 (Oct. 2018), 64–84. doi:10.1515/popets-2018-0032

[67] Lei Xu, Leqian Zheng, Chengzhi Xu, Xingliang Yuan, and Cong Wang. 2023. Leakage-Abuse Attacks Against Forward and Backward Private Searchable Symmetric Encryption. In *ACM CCS 2023: 30th Conference on Computer and Communications Security*, Weizhi Meng, Christian Damsgaard Jensen, Cas Cremers, and Engin Kirda (Eds.). ACM Press, Copenhagen, Denmark, 3003–3017. doi:10.1145/3576915.3623085

[68] Leqian Zheng, Lei Xu, Cong Wang, Sheng Wang, Yuke Hu, Zhan Qin, Feifei Li, and Kui Ren. 2024. SWAT: A System-Wide Approach to Tunable Leakage Mitigation in Encrypted Data Stores. *Proc. VLDB Endow.* 17, 10, 2445–2458. doi:10.14778/3675034.3675038

[69] Mingxun Zhou, Elaine Shi, T.-H. Hubert Chan, and Shir Maimon. 2023. A Theory of Composition for Differential Obliviousness. In *Advances in Cryptology – EUROCRYPT 2023, Part III (Lecture Notes in Computer Science, Vol. 14006)*, Carmit Hazay and Martijn Stam (Eds.). Springer, Cham, Switzerland, Lyon, France, 3–34. doi:10.1007/978-3-031-30620-4_1

[70] Mingxun Zhou, Mengshi Zhao, T.-H. Hubert Chan, and Elaine Shi. 2024. Advanced Composition Theorems for Differential Obliviousness. In *ITCS 2024: 15th Innovations in Theoretical Computer Science Conference*, Venkatesan Guruswami (Ed.), Vol. 287. Leibniz International Proceedings in Informatics (LIPIcs), Berkeley, CA, USA, 103:1–103:24. doi:10.4230/LIPIcs.ITCS.2024.103

A Differential Privacy (Extended)

Laplace Mechanism. We now briefly review the Laplace mechanism that achieves ϵ-DP [23]. This mechanism is suited for a general function $f(\cdot)$. The only requirement is that $f(\cdot)$ has bounded *sensitivity*. By sensitivity, we refer to the maximum distance (measured by some metric) in f's output when its inputs (e.g., a database) change by a single record. We denote this sensitivity using Δf. For simplicity, we focus on the ℓ_1 distance and the corresponding ℓ_1-sensitivity.

The Laplace mechanism $\mathcal{M}_L$ operates in the following manner: for each query, it first evaluates $f(\cdot)$ on the input, and then adds noise to each coordinate of the output. This noise is drawn from a Laplace distribution. The scale of the Laplace distribution is determined by the ℓ_1-sensitivity Δf and the privacy budget ϵ.

Let b denote the scale of the Laplace distribution. We can write its probability density function as follows,

$$\text{Lap}(x \mid b) = \frac{1}{2b} \exp\left(-\frac{x}{b}\right).$$

We also use $\text{Lap}(b)$ to represent $\text{Lap}(x \mid b)$. Then [22, 23] show that if we set $b = \frac{\Delta f}{\epsilon}$, $\mathcal{M}_L$ is ϵ-DP.

To better understand the concept, let us look at an additional example besides the count example provided in the introduction. Consider a function, f that computes the histogram of a database. The function output, a histogram, can be seen as a vector of reals. It becomes clear that its ℓ_1-sensitivity $\Delta f = 2$, as one individual record can influence, at most, two coordinates with an offset of 1. Then the Laplace mechanism $\mathcal{M}_L$ outputs the true histogram, with added noise in every coordinate. It can be shown that $\mathcal{M}_L$ is ϵ-DP, if the noise is drawn independently from $\text{Lap}(2/\epsilon)$.

Composition Theorem. As mentioned before, DP is composable. Here, we provide a basic composition theorem. If one mechanism $\mathcal{M}_1$ is (ϵ_1, δ_1)-DP for some $\epsilon_1 > 0, \delta_1 \geq 0$, and another mechanism $\mathcal{M}_2$ is (ϵ_2, δ_2)-DP for $\epsilon_1 > 0, \delta_1 \geq 0$, then the composition of the two, $\mathcal{M}_1 \circ \mathcal{M}_2$ is $(\epsilon_1 + \epsilon_2, \delta_1 + \delta_2)$-DP [22].[17]

B Structured Encryption (STE)

We use the syntax of structured encryption (STE) [25, 36] in our analysis because it is general enough, covering data structures we are interested in: range-query structure. We recall the formal definitions here.

Spaces and Data Structures. We use $\mathbb{L}$ to denote the *label space*, the domain of searchable values in the data structure (i.e., database). This is also known as *keyword space*. We use $\mathbb{V}$ to denote the (non-searchable) *value space*, e.g., the associated record content. Even though STE considers arbitrary data structures, our focus is on data structures of specific type, specifically, data structure $\text{ds} \subseteq \mathbb{L} \times \mathbb{V}$. The operation type space $\mathbb{O}$ is also fixed by the data structure, e.g., $\{\texttt{Insert}, \texttt{Find}\}$.

Map. Given label space $\mathbb{L}$, value space $\mathbb{V}$, a *map* is an unordered set of pairs denoted by $\text{ds} = \{(\ell_i, v_i)\}_{i \in \mathbb{N}}$, where $\ell_i \in \mathbb{L}$ and $v_i \in \mathbb{V}$. We use $\text{ds}[\ell_i]$ to denote ℓ_i's associated value, i.e., $\text{ds}[\ell_i] = v_i$.

Syntax. Structured encryption STE is defined by the label space $\mathbb{L}$, value space $\mathbb{V}$, operation type space $\mathbb{O}$, functionality $\mathcal{F}$, two algorithms, and one two-party protocol:

- $(\text{sk}, \text{eds}) \leftarrow\$ \text{Setup}(1^\lambda, z, \text{ds})$: is a randomized algorithm run by client **C** that takes as input a security parameter $\lambda \in \mathbb{N}$, auxiliary information $z \in \{0,1\}^*$ (e.g., the maximum size of the data structure), data structure $\text{ds} \subseteq \mathbb{L} \times \mathbb{V}$ with functionality $\mathcal{F}$, and outputs secret key sk and encrypted data structure eds.
- $(v^*, \text{eds}^*) \leftarrow [\text{Access}_{\mathbf{C}}(\text{sk}, \text{op}, \ell, \ell', v), \text{Access}_{\mathbf{S}}(\text{eds})]$: is a two-party protocol executed between client **C** and server **S**, where **C** inputs secret key sk, operation type $\text{op} \in \mathbb{O}$, label $\ell, \ell' \in \mathbb{L}$ (indicating a range $[\ell, \ell']$; for point queries, setting $\ell' \leftarrow \ell$), and a value $v \in \mathbb{V}$; server inputs eds; at the end of the protocol, **C** receives $v^* \subseteq \mathbb{V}$, and **S** updates the encrypted data structure to eds^*.
- $\text{ds} \leftarrow \text{Dec}(\text{sk}, \text{eds})$: is a deterministic algorithm that takes as input the secret key sk and the encrypted data structure eds, and outputs ds.

Reactive Functionality. We formally describe the basic operations associated with the data structure that permits accessing and adding the data. Following the ORAM literature, we say that the functionality is *reactive* if it holds an internal state between the executions. We only define the reactive functionality $\mathcal{F}_{\text{RQ}}$ for a range-query structure here as it is sufficient for our analysis. We set label space $\mathbb{L}$ to be integers; value space $\mathbb{V} = \{\{0,1\}^c\}^*$ (c is some constant).

Definition 4 ($\mathcal{F}_{\text{RQ}}$)**.** Given label space $\mathbb{L}$, value space $\mathbb{V}$, and operation type space $\mathbb{O} = \{\texttt{Insert}, \texttt{Find}\}$, we define reactive functionality $\mathcal{F}_{\text{RQ}}$, for every data structure $\text{ds} \subseteq \mathbb{L} \times \mathbb{V}$, $\mathcal{F}_{\text{RQ}}(\text{ds}, \text{op}, \ell, \ell', v)$: where range-query data structure ds is stored in the state,[18] $\text{op} \in \mathbb{O}$, $\ell, \ell' \in \mathbb{L}$, and $v \in \mathbb{V}$,

(1) If $\text{op} = \texttt{Insert}$, if ℓ is in ds, then $v^* \leftarrow \text{ds}[\ell]$, update $\text{ds}[\ell]$ to v^* appended with v; otherwise $\text{ds}[\ell] \leftarrow (v)$.
(2) If $\text{op} = \texttt{Find}$, if $\ell \leq \ell'$, then return a list of v^*, where $v^* \leftarrow \text{ds}[\ell^*]$, for every $\ell \leq \ell^* \leq \ell'$ and ℓ^* is in ds; otherwise $v^* \leftarrow \perp$.
(3) Output (v^*, ds).

Remark 4. For point queries, ℓ' is set to be equal to ℓ. When performing either point or range queries using Find, input v is set to $\perp$ in Definition 4.

Correctness. We intentionally omit the correctness requirement to make STE definition general. The correctness requirement can be instantiated differently depending on the scheme.

STE Security Definition. We present the standard STE simulation-based adaptive security in our style [25, 36]. The adversary can choose the data structure (with data), adaptively make updates and queries, and is given the encrypted data structure. This models many practical settings where the attacker does not fully corrupt the system but still can influence and observe the communication. As usual, we use an abstract leakage profile $\mathcal{L}$ that specifies the information that the protocol leaks, which can be accessed by the simulator in the security definition. We say that an STE scheme is adaptively $\mathcal{L}$-secure if no efficient adversary can distinguish the transcripts from the honest execution of the protocol in the real world and the simulated ones in the ideal world with non-negligible probability.

Definition 5 (Adaptive Security for STE)**.** Given label space $\mathbb{L}$, value space $\mathbb{V}$, and operation type space $\mathbb{O}$, let Π be a STE protocol with functionality $\mathcal{F}$. Let $\mathcal{L}_\Pi = (\mathcal{L}_{\text{Setup}}, \mathcal{L}_{\text{Access}})$ be the leakage profile describing leakage of Π's algorithms. Let $\lambda \in \mathbb{N}$ be the

[17] We omit the discussion on the advanced composition theorem.

[18] Without loss of generality, we use map as the abstraction of the state.

$\mathbf{Real}_{\Pi,\mathcal{A},q}(1^\lambda, z)$:

1: Given $1^\lambda, z$, adversary $\mathcal{A}$ outputs a data structure $\mathsf{ds}_1 \subseteq \mathbb{L} \times \mathbb{V}$.
2: $(K, \mathsf{eds}_1) \leftarrow\$ \mathsf{Setup}(1^\lambda, z, \mathsf{ds}_1)$.
3: $\mathcal{A}$ is given eds_1.
4: $\mathcal{A}$ adaptively makes q queries in any order,
5: **for** all $t \in [q]$ **do**
6: $\quad \mathsf{Access}(\mathsf{op}_t, \ell_t, \ell'_t, v_t)$:
7: $\quad$ where $\mathsf{op}_t \in \mathbb{O}, \ell_t, \ell'_t \in \mathbb{L}, v_t \in \mathbb{V}$,
8: $\quad$ returns to $\mathcal{A}$ the transcript of honest execution
9: $\quad [\mathsf{Access}_{\mathbf{C}}(K, \mathsf{op}_t, \ell_t, \ell'_t, v_t), \mathsf{Access}_{\mathbf{S}}(\mathsf{eds}_t)]$,
10: $\quad$ and server $\mathbf{S}$'s output eds_{t+1}.
11: **end for**
12: Finally, $\mathcal{A}$ outputs bit b.
13: The experiment returns the same bit b.

$\mathbf{Ideal}_{\Pi,\mathcal{A},\mathcal{S},\mathcal{F},q}(1^\lambda, z)$:

1: Given $1^\lambda, z$, adversary $\mathcal{A}$ outputs a data structure $\mathsf{ds}_1 \subseteq \mathbb{L} \times \mathbb{V}$.
2: Given $1^\lambda, z, \mathcal{L}_{\mathsf{Setup}}(\mathsf{ds}_1)$, simulator $\mathcal{S}$ outputs encrypted data structure eds_1 and sends it to $\mathcal{A}$.
3: Let $\mathbf{S}$ be an honest server and given eds_1.
4: $\mathcal{A}$ adaptively makes q queries in any order,
5: **for** all $t \in [q]$ **do**
6: $\quad \mathsf{Access}(\mathsf{op}_t, \ell_t, \ell'_t, v_t)$:
7: $\quad$ where $\mathsf{op}_t \in \mathbb{O}, \ell_t, \ell'_t \in \mathbb{L}, v_t \in \mathbb{V}$,
8: $\quad$ returns to $\mathcal{A}$ the transcript of
9: $\quad [\mathcal{S}(\mathcal{L}_{\mathsf{Access}}(\mathsf{ds}_t, \mathsf{op}_t, \ell_t, \ell'_t, v_t)), \mathsf{Access}_{\mathbf{S}}(\mathsf{eds}_t)]$,
10: $\quad$ and server $\mathbf{S}$'s output eds_{t+1};
11: $\quad \mathsf{ds}_{t+1} \leftarrow \mathcal{F}(\mathsf{ds}_t, \mathsf{op}_t, \ell_t, \ell'_t, v_t)$.
12: **end for**
13: Finally, $\mathcal{A}$ outputs bit b.
14: The experiment returns the same bit b.

Figure 6: Experiments for Defining STE Adaptive Security.

security parameter. Consider the probabilistic experiments defined in Figure 6.

We say that Π is adaptively $\mathcal{L}_\Pi$-secure if for all $z \in \{0,1\}^*$, there exists a PPT simulator $\mathcal{S}$ such that for all PPT adversaries $\mathcal{A}$, all q which is a polynomial (in λ), the following is negligible (in λ):

$$\left| \Pr\left[\mathbf{Real}_{\Pi,\mathcal{A},q}(1^\lambda, z) = 1 \right] - \Pr\left[\mathbf{Ideal}_{\Pi,\mathcal{A},\mathcal{S},\mathcal{F},q}(1^\lambda, z) = 1 \right] \right|.$$

C Differentially Private Access and Proofs

C.1 The Non-Adaptive Definition

Definition 6 (Non-Adaptive Differentially Private Access)**.** Given label space $\mathbb{L}$, value space $\mathbb{V}$, and operation type space $\mathbb{O}$, let Π be an STE protocol with functionality $\mathcal{F}$. Let $\lambda \in \mathbb{N}$ be the security parameter. For $b \in \{0,1\}$, we define the security experiment in Figure 7.

$\mathbf{Exp}^{\text{dp-access},b}_{\Pi,\mathcal{A},s}(1^\lambda, z, \mathsf{ds}^0, \mathsf{ds}^1, \mathbf{q})$:

1: $(\mathsf{sk}, \mathsf{eds}_1) \leftarrow\$ \mathsf{Setup}(1^\lambda, z, \mathsf{ds}^b)$.
2: $\mathcal{A}$ is given ds^0, ds^1, $\mathbf{q}$, λ, z and eds_1.
3: **for** all $i \in [s]$ **do**
4: $\quad \mathsf{Access}(\mathtt{Find}, \mathbf{q}[i], \perp)$:
5: $\quad$ returns to $\mathcal{A}$ the transcript of honest execution of
6: $\quad [\mathsf{Access}_{\mathbf{C}}(\mathsf{sk}, \mathtt{Find}, \mathbf{q}[i], \perp), \mathsf{Access}_{\mathbf{S}}(\mathsf{eds}_i)]$ and
7: $\quad$ Server $\mathbf{S}$'s output eds_{i+1}.
8: **end for**
9: Finally, $\mathcal{A}$ outputs bit b'.
10: The experiment returns the same bit b'.

Figure 7: Non-Adaptive Differentially Private Access (DP-Access) Experiment.

We say that Π has non-adaptive (ϵ, δ)-DP access leakage if for all $z \in \{0,1\}^*$, all neighboring ds_0 and ds_1 over $\mathbb{L} \times \mathbb{V}$, all find queries $\mathbf{q} \subseteq \mathbb{L} \times \mathbb{L}$ of length s, which is a polynomial (in λ), all PPT adversary $\mathcal{A}$, all $b' \in \{0,1\}$,

$$\Pr\left[\mathbf{Exp}^{\text{dp-access},b}_{\Pi,\mathcal{A},s}(1^\lambda, z, \mathsf{ds}^0, \mathsf{ds}^1, \mathbf{q}) = b' \right] \leq$$
$$e^\epsilon \cdot \Pr\left[\mathbf{Exp}^{\text{dp-access},1-b}_{\Pi,\mathcal{A},s}(1^\lambda, z, \mathsf{ds}^0, \mathsf{ds}^1, \mathbf{q}) = b' \right] + \delta.$$

C.2 Proof for Non-Adaptive DP-Access of ϵ-OPE-Range

Overview. Chowdhury et al. [16] propose an ϵ-OPE encryption scheme. It is an extension of order preserving encryption (OPE) with differential privacy. We refer to the encrypted range query scheme using ϵ-OPE ciphertexts proposed by them as ϵ-OPE-Range. They capture the security guarantee of ϵ-OPE ciphertexts with a game-based security definition called ϵ-IND-FA-OCPA (cf. Definition 4.2 [16], which we recall in Definition 8). This hybrid security definition combines ϵ-dDP, a distance-based differential privacy notion (i.e., defining the neighboring condition based on a distance metric), and IND-FA-OCPA, a security notion for OPE [38]. The term "FA" refers to "frequency-analyzing", meaning that the ciphertexts do not reveal the equality relation among the ciphertexts. "OCPA" stands for "ordered chosen plaintext attacks", tailored for the indistinguishability-based security game for ciphertexts that preserve the plaintext order. The IND-FA-OCPA notion addresses these two properties in the indistinguishability-based game by requiring the two query sequences of equal length sharing at least a *common randomized order*. We omit the formal definition of randomized order here. Instead, we provide a simple example to illustrate: the tuple $(1,2,2,3)$ has two randomized (increasing) orders, $(1,2,3,4)$ and $(1,3,2,4)$. The randomness lies in the equality relation, as the second and third elements are both equal to 2.

To show that ϵ-OPE-Range built from ϵ-OPE ciphertexts, proven secure in ϵ-IND-FA-OCPA, also satisfies our DP access notion, we adopt the standard reduction approach. Suppose there is a PPT adversary $\mathcal{A}$ that breaks the game-based definition ϵ-IND-FA-OCPA. Then we show we can construct an adversary $\mathcal{A}'$ using $\mathcal{A}$ as a subroutine to break DP access. Before moving to the proof, we

first recall the security definition ϵ-IND-FA-OCPA. We begin by describing the notation.

Notation. The label space (or domain) is denoted as $\mathcal{X}$, which has a total ordering.[19] Then this domain can be partitioned into $(\mathcal{X}_i)_{i\in[k]}$ following this ordering, for some k. As an example, consider the discrete range $[1, 100]$. For 4-partitions, a partition of equal size (except for the first range) is a set of discrete ranges $\{[1, 25], (25, 50], (50, 75], (75, 100]\}$.

The partition universe (i.e., all possible partitions) over $\mathcal{X}$ is denoted by $\hat{\mathcal{X}}$. Then any partition over $\mathcal{X}$, denoted by $\mathcal{P}$ is in $\hat{\mathcal{X}}$. This partition notation $\mathcal{P}$ is also used in work [16] to denote the partition algorithm $\mathcal{P}(\cdot)$. It takes as input an element from $\mathcal{X}$ and outputs the partition index. This partition index is called an "encoding" that lies in some output domain $\mathcal{O}$. More specifically, $\mathcal{P}(\cdot) : \mathcal{X} \to \mathcal{O}$. Continuing with the example of the 4-partition set we mentioned before, let (o_1, o_2, o_3, o_4) be the corresponding encoding of the partition indices. Then, $\mathcal{P}(40) = o_2$.

Randomized Encoding Mechanism OPϵc **and** ϵ-dLDP**.** The partition algorithm we have discussed so far is deterministic, meaning that there is no noise to guarantee DP yet. The authors' idea is to add DP to the partition algorithm by introducing a randomized encoding mechanism called OPϵc [16]. This randomized OPϵc takes as input $(x, \mathcal{P}, \epsilon)$, where $x \in \mathcal{X}$, partition $\mathcal{P} \in \hat{\mathcal{X}}$, privacy budget $\epsilon > 0$, and outputs an encoding in $\mathcal{O}$. For example, OP$\epsilon c(40, \mathcal{P}, \epsilon)$ can output o_2 and possibly o_1 or o_3 as well. They prove their OPϵc construction satisfies distance-based local differential privacy (ϵ-dLDP) [16]. We briefly recall the definition ϵ-dLDP here and discuss its counterpart distance-based central differential privacy (ϵ-dDP).

Definition 7 (Distance-based Local Differential Privacy, ϵ-dLDP [16])**.** A randomized algorithm $\mathcal{M} : \mathcal{X} \to \mathcal{Y}$ is ϵ-distance based locally differentially private (ϵ-dLDP), if for any pair of private values $x, x' \in \mathcal{X}$ and any subset of output $T \subseteq \mathcal{Y}$,

$$\Pr[\mathcal{M}(x) \in \mathcal{T}] \leq e^{\epsilon|x-x'|} \cdot \Pr[\mathcal{M}(x') \in \mathcal{T}].$$

Comparison between ϵ-dLDP **and** ϵ-dDP **[16].** Both ϵ-dLDP and ϵ-dDP define the neighboring condition using a distance metric (e.g., ℓ_1 norm). Their difference lies in the input to the DP randomized mechanism $\mathcal{M}$. In ϵ-dLDP, $\mathcal{M}$ takes as input a message. Naturally, the neighboring condition is defined based on the distance between two messages. In comparison, in ϵ-dDP, $\mathcal{M}$ takes as input a sequence of messages. The neighboring condition is adjusted accordingly: the two sequences differ in only one message, and the distance metric is applied to those two messages. We refer to work [16] for full details. Later, we show how we can instantiate our neighboring condition (cf. Definition 2) to be equivalent to their neighboring definition in our proof.

OPE Syntax. Because the ϵ-IND-FA-OCPA notion is intertwined with both DP and OPE notions and related syntax, we also recall the OPE syntax here [38] [16]. An order-preserving encryption (OPE) scheme contains three algorithms (OPE.KeyGen, OPE.Enc, OPE.Dec) over some message domain $\mathcal{X}$ as follows,

- $\mathsf{sk} \leftarrow\$ \mathsf{OPE.KeyGen}(1^\kappa)$: takes as input a security parameter $\kappa \in \mathbb{N}$, and outputs a random state $\mathsf{sk} \in \{0,1\}^*$.
- $(\mathsf{sk}', y) \leftarrow\$ \mathsf{OPE.Enc}(\mathsf{sk}, x, \Gamma)$: takes as input a secret state sk, a message $x \in \mathcal{X}$, some (optional) randomized order Γ, and outputs an updated state sk' and a ciphertext y.
- $x \leftarrow \mathsf{OPE.Dec}(\mathsf{sk}, y)$: takes as input a secret state sk, ciphertext y, and outputs the message x.

We abuse the notation OPE.Enc to denote the encryption algorithm for a list of messages. For messages $X \in \mathcal{X}^n$, the notation $(\mathsf{sk}', Y) \leftarrow\$ \mathsf{OPE.Enc}(\mathsf{sk}, X, \Gamma)$ denotes the following operations: $\mathsf{sk}_1 \leftarrow \mathsf{sk}$; for every i from 1 to n: $(\mathsf{sk}_{i+1}, y_1) \leftarrow\$ \mathsf{OPE.Enc}(\mathsf{sk}_i, X[i], \Gamma)$; $Y \leftarrow (y_1, y_2, \ldots, y_n)$; output (sk_{n+1}, Y).

We omit the formal correctness requirement. As OPE ciphertexts preserve the plaintext order, OPE naturally supports the following two functionalities:

- $\{\leq, \geq\} \leftarrow \mathsf{OPE.Compare}(y_1, y_2)$: takes as input ciphertexts y_1, y_2, and outputs an inequality relation in $\{\leq, \geq\}$. This relation applies to the underlying messages of y_1 and y_2. We write $y_1 \leq y_2$ or $y_1 \geq y_2$ as a shorthand.
- $(y'_1, y'_2, \ldots, y'_n) \leftarrow \mathsf{OPE.Sort}(y_1, y_2, \ldots, y_n)$: takes as input a list of ciphertexts $y_1, y_2, \ldots, y_n$, and outputs a sorted list of ciphertexts in ascending order, $y'_1 \leq y'_2 \leq \cdots \leq y'_n$.

We now have enough background, and we can finally describe their generic construction OPϵ [16], which we refer to as ϵ-OPE.

Generic Construction ϵ-OPE **(**OP$_\epsilon$**) [16].** Given an order-preserving scheme OPE that is IND-FA-OCPA, an OPϵc encoding scheme that is ϵ-dLDP, then ϵ-OPE = (ϵ-OPE.KeyGen, ϵ-OPE.Enc, ϵ-OPE.Dec) can be constructed generically as follows,

- $\mathsf{sk} \leftarrow\$ \epsilon\text{-}\mathsf{OPE.KeyGen}(1^\kappa)$: run $\mathsf{sk} \leftarrow\$ \mathsf{OPE.KeyGen}(1^\kappa)$ and then outputs sk.
- $(\mathsf{sk}', y) \leftarrow\$ \epsilon\text{-}\mathsf{OPE.Enc}(\mathsf{sk}, x, \Gamma, \mathcal{P}, \epsilon)$: takes as input the secret key sk, a message $x \in \mathcal{X}$, an order Γ, a partition $\mathcal{P}$ from the partition universe $\hat{\mathcal{X}}$, some privacy budget $\epsilon > 0$; run $\tilde{o} \leftarrow\$ \mathsf{OP}\epsilon c(x, \mathcal{P}, \frac{\epsilon}{2})$; $(\mathsf{sk}', y) \leftarrow\$ \mathsf{OPE.Enc}(\mathsf{sk}, \tilde{o}, \Gamma)$, and outputs (sk', y).
- $\tilde{o} \leftarrow \epsilon\text{-}\mathsf{OPE.Dec}(\mathsf{sk}, y)$: run $\tilde{o} \leftarrow \mathsf{OPE.Dec}(\mathsf{sk}, y)$, and outputs $\tilde{o}$.

Remark 5 (On Decryption). It is clear that ϵ-OPE does not support retrieving the original message x from the ciphertext y. This issue can be resolved by augmenting ϵ-OPE with an IND-CPA (or IND-CCA) blockcipher-based encryption scheme $\mathcal{E}$. Essentially, for each ϵ-OPE ciphertext, an additional ciphertext is stored, encrypting the label x and associated value v using $\mathcal{E}$. We refer to work [16] for details.

ϵ-OPE **Security.** Chowdhury et al. [16] show that ϵ-OPE ciphertexts are $\frac{\epsilon}{2}$-dDP and satisfy ϵ-IND-FA-OCPA that we recall in Definition 8.

Encrypted Range Queries ϵ-OPE-Range **from** ϵ-OPE**.** The authors [16] also demonstrate how to use ϵ-OPE for encrypted range queries. We refer to their construction as ϵ-OPE-Range. It is important to note that, ϵ-OPE-Range implements the STE syntax. We use ϵ-OPE-Range = (ϵ-OPE-Range.Setup, ϵ-OPE-Range.Access,

[19] We simplify this case from partial order. Total ordering is sufficient for practical cases, as searchable labels are usually integers and floats.

ϵ-OPE-Range.Dec) to denote their setup algorithm, query algorithm, and decryption algorithm respectively. For more details, we refer to their paper [16].

Definition 8 (ϵ-IND-FA-OCPA [16])**.** An ϵ-OPE encryption scheme Π has indistinguishable ciphertexts under frequency-analyzing ϵ-dDP ordered chosen plaintext attacks if for every PPT adversary $\mathcal{A}$, and security parameter κ :

$$\Pr[\mathbf{Exp}^{\epsilon\text{-ind-fa-ocpa}}_{\mathcal{A},\Pi}(\kappa, b_1, b_2) = (c_1, c_2)] \leq$$
$$e^{t\epsilon} \cdot \Pr[\mathbf{Exp}^{\epsilon\text{-ind-fa-ocpa}}_{\mathcal{A},\Pi}(\kappa, b_1', b_2') = (c_1, c_2)] + \mathsf{negl}(\kappa),$$

where $b_1, b_2, b_1', b_2', c_1, c_2 \in \{0,1\}$ and $\mathbf{Exp}^{\text{FA-OCPA}\epsilon}_{\mathcal{A},\Pi}(\kappa, b_1, b_2)$ is the random variable indicating the adversary $\mathcal{A}$'s output for the security game in Figure 8.

$\mathbf{Exp}^{\epsilon\text{-ind-fa-ocpa}}_{\Pi,\mathcal{A}}(\kappa, b_1, b_2)$:

$(X_{00}, X_{01}, X_{10}, X_{11}) \leftarrow\$\ \mathcal{A}$ where
  $X_{00}, X_{01} \in \mathcal{X}_0^n$ and $X_{10}, X_{11} \in \mathcal{X}_1^n$;
  $\mathcal{P}_0(X_{00})$ and $\mathcal{P}_1(X_{10})$ have at least one common randomized order where $\mathcal{P}_0 \in \hat{\mathcal{X}}_0$ and $\mathcal{P}_1 \in \hat{\mathcal{X}}_1$;
  $\{X_{00}, X_{01}\}$ and $\{X_{10}, X_{11}\}$ are t-adjacent.
$\mathsf{sk} \leftarrow\$\ \epsilon\text{-OPE.KeyGen}(1^\kappa)$
Compute $\widetilde{O}_0 \leftarrow\$\ \mathsf{OP}\epsilon c\left(X_{00}, \mathcal{P}_0, \frac{\epsilon}{2}\right)$ and $\widetilde{O}_1 \leftarrow\$\ \mathsf{OP}\epsilon c\left(X_{10}, \mathcal{P}_1, \frac{\epsilon}{2}\right)$.
if $\widetilde{O}_0$ and $\widetilde{O}_1$ do not have any common randomized order **then**
  return $\perp$
else
  Select two uniform bits b_1 and b_2 and a randomized order Γ^* common to both $\widetilde{O}_0$ and $\widetilde{O}_1$.
end if
if $b_2 = 0$ **then**
  ▷ *Equivalent to running* ϵ-OPE.Enc$(\mathsf{sk}, X_{b_1 0}, \Gamma^*, \mathcal{P}_{b_1}, \epsilon)$
  Run ϵ-OPE.Enc$(\mathsf{sk}, \widetilde{O}_{b_1}, \Gamma^*, \mathcal{O}_{b_1}, \infty)$:
    ▷ $\mathcal{O}_{b_1}$ *is the encoding domain for* $\mathcal{P}_{b_1}$
    Compute $\widetilde{O}'_{b_1} \leftarrow\$\ \mathsf{OP}\epsilon c(\widetilde{O}_{b_1}, \mathcal{O}_{b_1}, \infty)$
      ▷ $\widetilde{O}'_{b_1} = \widetilde{O}_{b_1}$ *(See Definition 4.2 [16])*
    $(\mathsf{sk}', Y_{b_1 b_2}) \leftarrow\$\ \mathsf{OPE.Enc}(\mathsf{sk}, \widetilde{O}'_{b_1}, \Gamma^*)$
else
  Run ϵ-OPE.Enc$(\mathsf{sk}, X_{b_1 1}, \Gamma^*, \mathcal{P}_{b_1}, \epsilon)$:
    Compute $\widetilde{O}'_{b_1} \leftarrow\$\ \mathsf{OP}\epsilon c(X_{b_1 1}, \mathcal{P}_{b_1}, \frac{\epsilon}{2})$
    $(\mathsf{sk}', Y_{b_1 b_2}) \leftarrow\$\ \mathsf{OPE.Enc}(\mathsf{sk}, \widetilde{O}'_{b_1}, \Gamma^*)$
end if
$(c_1, c_2) \leftarrow\$\ \mathcal{A}(Y_{b_1 b_2})$, where $c_1(c_2)$ is $\mathcal{A}$'s guess for $b_1(b_2)$
$\mathcal{A}$ is said to win this experiment if $b_1 = c_1$ or $b_2 = c_2$
return (c_1, c_2)

Figure 8: Experiment ϵ-IND-FA-OCPA.

Theorem C.1. *If ϵ-OPE-Range is built from ϵ-OPE ciphertexts satisfying ϵ-IND-FA-OCPA, then ϵ-OPE-Range also satisfies our non-adaptive* DP-Access *notion.*

Proof. We start by briefly describing their security experiment ϵ-IND-FA-OCPA (Figure 8). We have rewritten some of the notations in their definition for clarity. Intuitively, this security experiment, ϵ-IND-FA-OCPA, captures the property that the ciphertexts of four message tuples $(X_{b_1 b_2})_{b_1, b_2 \in \{0,1\}}$ are "indistinguishable" (including in the weak DP-indistinguishable case). The challenge bits are b_1 and b_2. The adversary, denoted as $\mathcal{A}$, tries to guess either b_1 or b_2 when given the ciphertext of X_{b_1, b_2} encrypted using ϵ-OPE. It wins the experiment if its guess for either b_1 or b_2 is correct.

For illustration purposes, we assume the message domain $\mathcal{X}_1 = \mathcal{X}_2$ are denoted by $\mathcal{X}$, and the partitions $\mathcal{P}_1 = \mathcal{P}_2$ are denoted by $\mathcal{P}$. The security game is initialized as follows. The adversary outputs two pairs of messages (X_{00}, X_{01}), (X_{10}, X_{11}), each of length n, with two constraints. The first constraint is that (X_{00}, X_{01}) and (X_{10}, X_{11}) are both t-adjacent. This is a requirement preparing for the challenge on ϵ-dDP. The second constraint is that $\mathcal{P}(X_{00})$ and $\mathcal{P}(X_{10})$ share at least one common randomized order, preparing for the challenge on IND-FA-OCPA.

This hybrid experiment needs to somehow combine the two challenges. This is achieved by first applying the randomized encoding mechanism OPϵc to add some DP noise to the partition output (i.e., encoding) of the message tuple, and then applying OPE. Chowdhury et al. have proved that to ensure the ϵ-OPE scheme satisfies ϵ-IND-FA-OCPA, it should have an internal call to OPϵc with $\frac{\epsilon}{2}$ in the encryption algorithm. Let $\widetilde{O}_0 \leftarrow\$\ \mathsf{OP}\epsilon c(X_{00}, \mathcal{P}, \frac{\epsilon}{2})$ and $\widetilde{O}_1 \leftarrow\$\ \mathsf{OP}\epsilon c(X_{10}, \mathcal{P}, \frac{\epsilon}{2})$. Since the OP$\epsilon c$ output is noisy, it is possible that $\widetilde{O}_0$ and $\widetilde{O}_1$ no longer share at least one common randomized order. If that is the case, they no longer satisfy the constraint for the IND-FA-OCPA game; the ϵ-IND-FA-OCPA experiment aborts and re-draws the random bits b_1 and b_2. It only proceeds when $\widetilde{O}_0$ and $\widetilde{O}_1$ share at least one common randomized order. This artificial technicality ensures the security definition is meaningful irrespective of the correctness of OPϵc.[20]

Now consider the game variations based on the challenge bits. At a high-level, the input to OPE.Enc is either $\widetilde{O}_{b_1}$ for $b_2 = 0$ or output from $\mathsf{OP}\epsilon c(X_{b_1 1}, \mathcal{P}, \frac{\epsilon}{2})$ for $b_2 = 1$. Consider a simple example to help with understanding. Because both (X_{00}, X_{01}) and (X_{10}, X_{11}) are t-adjacent, their output encodings from OPϵc are likely to be the same. These encodings are input to OPE.Enc, so their ciphertexts are "indistinguishable". This pairwise ciphertext indistinguishability is further augmented by IND-FA-OCPA for (X_{00}, X_{10}), as they respective encodings are taken as input to OPE.Enc for the $b_2 = 0$ case. Finally, the four ciphertexts of $X_{00}, X_{01}, X_{10}, X_{11}$ are pairwise indistinguishable.

As we are now familiar with the ϵ-IND-FA-OCPA security game, let us proceed with our proof. We provide a proof by contradiction.

First, we define the neighboring condition in our DP-Access notion (cf. Definition 3). We instantiate the abstract distance function in the neighboring condition as follows: let $\mathtt{dist}(\mathsf{ds}_0, \mathsf{ds}_1) = 1$ if and only if ds_0 and ds_1 differ in exactly one pair of label and value, and their difference in the label pairs is exactly 1 under the same distance metric of ϵ-IND-FA-OCPA [16]. Similar to ϵ-IND-FA-OCPA, for every two databases, we say that they are t-adjacent if $\mathtt{dist}(\mathsf{ds}_0, \mathsf{ds}_1) \leq t$.

Next, we choose security parameter $\lambda = \kappa$, and $\delta = \mathsf{negl}(\kappa)$.

Plugging these parameters into the DP-Access notion (cf. Definition 3), setting $b = 1$ and $b' = 1$, we obtain that for every PPT

[20] We note that this treatment does not necessarily mean it captures all related attacks.

adversary $\mathcal{A}$, every $z \in \{0,1\}^*$,

$$\Pr\left[\mathbf{Exp}^{\text{dp-access},1}_{\Pi,\mathcal{A},q}(1^\kappa, z, \mathsf{ds}_0, \mathsf{ds}_1) = 1\right] \leq e^{\epsilon} \cdot \Pr\left[\mathbf{Exp}^{\text{dp-access},0}_{\Pi,\mathcal{A},q}(1^\kappa, z, \mathsf{ds}_0, \mathsf{ds}_1) = 1\right] + \mathsf{negl}(\kappa).$$

For the t-adjacent variant of DP-Access, applying the DP composition theorem, we have the following inequality,

$$\Pr\left[\mathbf{Exp}^{\text{dp-access},1}_{\Pi,\mathcal{A},q}(1^\kappa, z, \mathsf{ds}_0, \mathsf{ds}_1) = 1\right] \leq e^{t\epsilon} \cdot \Pr\left[\mathbf{Exp}^{\text{dp-access},0}_{\Pi,\mathcal{A},q}(1^\kappa, z, \mathsf{ds}_0, \mathsf{ds}_1) = 1\right] + \mathsf{negl}(\kappa). \quad (1)$$

Our goal is to construct $\mathcal{B}$ for ϵ-IND-FA-OCPA if we have a PPT adversary $\mathcal{A}$ that breaks the t-adjacent DP-Access (Equation 1).

Without loss of generality, we consider the simplified case. That is, we let the message domain $\mathcal{X} = \mathcal{X}_0 = \mathcal{X}_1$ and the partition $\mathcal{P} = \mathcal{P}_0 = \mathcal{P}_1$. We construct $\mathcal{B}$ as follows: $\mathcal{B}$ chooses $(X_{00}, X_{01}, X_{10}, X_{11})$, where $X_{00}, X_{01}, X_{10}, X_{11} \in \mathcal{X}^n$ are t-adjacent, and $\mathcal{P}(X_{00})$ and $\mathcal{P}(X_{10})$ at positions 00 and 10 have at least one common randomized order, satisfying the requirement of the ϵ-IND-FA-OCPA game. Then follow the game ϵ-IND-FA-OCPA as in Figure 8 until $Y_{b_1 b_2}$ is computed.

As we do not need to pass any auxiliary information z to the DP-Access notion, we set $z = \perp$. We write $\mathcal{A}$'s probabilities against the non-adaptive t-adjacent DP-Access explicitly as follows,

$$\Pr\left[\mathbf{Exp}^{\text{dp-access},1}_{\mathcal{B},\mathcal{A},q}(1^\kappa, \perp, X_{00}, X_{01}) = 1\right] > e^{t\epsilon} \cdot \Pr\left[\mathbf{Exp}^{\text{dp-access},0}_{\mathcal{B},\mathcal{A},q}(1^\kappa, \perp, X_{00}, X_{01}) = 1\right] + \mathsf{negl}(\kappa).$$

This inequality holds for the $1/2$ of time, as for the other half, $Y_{b_1 b_2}$ is the ciphertext of one of the two plaintexts (X_{10}, X_{11}), and $\mathcal{A}$ is given an invalid ciphertext. Then the following inequality holds,

$$2 \cdot \Pr\left[\mathbf{Exp}^{\text{dp-access},1}_{\mathcal{B},\mathcal{A},q}(1^\kappa, \perp, X_{00}, X_{01}) = 1\right] > e^{t\epsilon} \cdot \Pr\left[\mathbf{Exp}^{\text{dp-access},0}_{\mathcal{B},\mathcal{A},q}(1^\kappa, \perp, X_{00}, X_{01}) = 1\right] + \mathsf{negl}(\kappa).$$

To win the ϵ-IND-FA-OCPA experiment, $\mathcal{A}$ needs to compute the plaintext of the challenge ciphertext $Y_{b_1 b_2}$ correctly, with a probability better than what is quantified by ϵ-IND-FA-OCPA security.

Let us now consider how $\mathcal{B}$ simulates the DP-Access game with $\mathcal{A}$. First $\mathcal{B}$ sample a random bit $b'_1 \leftarrow\$ \{0,1\}$. Then run $(1^\kappa, \perp, X_{b'_1 0}, X_{b'_1 1}, \perp)$ with $\mathcal{A}$ for the non-adaptive t-adjacent DP-Access game internally. The non-adaptive DP-Access experiment bit is same as b_2 from the ϵ-IND-FA-OCPA experiment. Essentially, this is a game by rewriting. After $\mathcal{B}$ receiving $Y_{b_1 b_2}$ from the ϵ-IND-FA-OCPA experiment. $\mathcal{B}$ simulates the setup algorithm and outputs $Y_{b_1 b_2}$ as the encrypted data structure eds_1; $\mathcal{A}$ is given $X_{00}, X_{01}, \kappa, \mathsf{eds}_1$ and $z = \perp$, $\mathbf{q} = \perp$. There is no Find queries.

Finally, at the end of the non-adaptive t-adjacent DP-Access experiment, $\mathcal{A}$ outputs a bit $b_2^{\mathcal{A}}$. Let $\mathcal{B}$ output $(b'_1, b_2^{\mathcal{A}})$ for the ϵ-IND-FA-OCPA experiment.

Now let us analyze why the above attack works. The core observation is that $Y_{b_1 b_2}$ is the ciphertext of $X_{b_1 b_2}$. We show how this is the case. First, the ϵ-IND-FA-OCPA game starts with computing encoding $\widetilde{O}_{b_1}$: $\widetilde{O}_0 \leftarrow\$ \mathsf{OP}\epsilon c(X_{00}, \mathcal{P}, \frac{\epsilon}{2})$ or $\widetilde{O}_1 \leftarrow\$ \mathsf{OP}\epsilon c(X_{01}, \mathcal{P}, \frac{\epsilon}{2})$. Consider when $b_2 = 0$, at Line 18, the output $\widetilde{O}'_{b_1} = \widetilde{O}_{b_1}$. Then, $\widetilde{O}'_{b_1}$ is input to OPE.Enc to compute $Y_{b_1 b_2}$. This implies that $Y_{b_1 0}$ is the encryption of $X_{b_1 0}$. For the $b_2 = 1$ case, consider messages at positions 01 and 11. At Line 22, when $b_1 = 0$, $\widetilde{O}'_0 \leftarrow\$ \mathsf{OP}\epsilon c(X_{01}, \mathcal{P}, \frac{\epsilon}{2})$; when $b_1 = 1$, $\widetilde{O}'_1 \leftarrow\$ \mathsf{OP}\epsilon c(X_{11}, \mathcal{P}, \frac{\epsilon}{2})$. This $\widetilde{O}'_{b_1}$ is further input to OPE.Enc to compute $Y_{b_1 1}$. Hence, $Y_{b_1 1}$ is the encryption of $X_{b_1 1}$. It follows that $Y_{b_1 b_2}$ is indeed the encryption of $X_{b_1 b_2}$. Then we can consider that b_1 chooses either (X_{00}, X_{01}) or (X_{10}, X_{11}) pairs. The bit b_2 determines it is the left or the right. The winning probability of $\mathcal{B}$ on b_2 now depends on the winning probability of $\mathcal{A}$ for DP-Access. Since we already assume that $\mathcal{A}$ breaks non-adaptive DP-Access and outputs $b_2^{\mathcal{A}}$, $\mathcal{B}$ can use $b_2^{\mathcal{A}}$ to break ϵ-IND-FA-OCPA.

Let us now analyze the exact probabilities of $\mathcal{B}$ breaking ϵ-IND-FA-OCPA. Let p_1 denote $\Pr\left[\mathbf{Exp}^{\text{dp-access},1}_{\mathcal{B},\mathcal{A},q}(1^\kappa, \perp, X_{b'_1 0}, X_{b'_1 1}, \perp) = 1\right]$, $\overline{p_1}$ denote $\Pr\left[\mathbf{Exp}^{\text{dp-access},1}_{\mathcal{B},\mathcal{A},q}(1^\kappa, \perp, X_{b'_1 0}, X_{b'_1 1}, \perp) = 0\right]$; then $\overline{p_1} = 1 - p_1$. Let p_0 denote $\Pr\left[\mathbf{Exp}^{\text{dp-access},0}_{\mathcal{B},\mathcal{A},q}(1^\kappa, \perp, X_{b'_1 0}, X_{b'_1 1}, \perp) = 0\right]$, $\overline{p_0}$ denote $\Pr\left[\mathbf{Exp}^{\text{dp-access},0}_{\mathcal{B},\mathcal{A},q}(1^\kappa, \perp, X_{b'_1 0}, X_{b'_1 1}, \perp) = 1\right]$, then $\overline{p_0} = 1 - p_0$.

Since $\mathcal{A}$ breaks non-adaptive t-adjacent DP-Access if $b'_1 = b_1$, and can be used to compute the answer for b_2. The following inequality holds,

$$\begin{aligned}
&\Pr\left[\mathbf{Exp}^{\text{dp-access},1}_{\mathcal{B},\mathcal{A},q}(1^\kappa, \perp, X_{b'_1 0}, X_{b'_1 1}, \perp) = 1\right] \\
&> e^{t\epsilon} \cdot \Pr\left[\mathbf{Exp}^{\text{dp-access},0}_{\mathcal{B},\mathcal{A},q}(1^\kappa, \perp, X_{b'_1 0}, X_{b'_1 1}, \perp) = 1\right] + \mathsf{negl}(\kappa) \\
&p_1 > e^{t\epsilon} \cdot \overline{p_0} + \mathsf{negl}(\kappa). \qquad (2)
\end{aligned}$$

Similarly, we have the following inequality by setting $b = 0$ and $b' = 0$ in non-adaptive t-adjacent DP-Access,

$$\begin{aligned}
&\Pr\left[\mathbf{Exp}^{\text{dp-access},0}_{\mathcal{B},\mathcal{A},q}(1^\kappa, \perp, X_{b'_1 0}, X_{b'_1 1}, \perp) = 0\right] > \\
&e^{t\epsilon} \cdot \Pr\left[\mathbf{Exp}^{\text{dp-access},1}_{\mathcal{B},\mathcal{A},q}(1^\kappa, \perp, X_{b'_1 0}, X_{b'_1 1}, \perp) = 0\right] + \mathsf{negl}(\kappa) \\
&p_0 > e^{t\epsilon} \cdot \overline{p_1} + \mathsf{negl}(\kappa). \qquad (3)
\end{aligned}$$

Let us compute the probability of correct guesses, output by $\mathcal{B}$ in the ϵ-IND-FA-OCPA game. Given $b_1, b_2 \leftarrow\$ \{0,1\}$, fixed by the game.

For correct guesses (b_1, b_2), we have the following,

$$\begin{aligned}
&\Pr[\mathbf{Exp}^{\epsilon\text{-ind-fa-ocpa}}_{\mathcal{B},\Pi}(\kappa, b_1, b_2) = (b_1, b_2)] \\
&= \Pr[b'_1 = b_1] \cdot \Pr[b_2^{\mathcal{A}} = b_2 \mid b'_1 = b_1] \\
&= \frac{1}{2} \cdot \Pr[b_2^{\mathcal{A}} = b_2 \mid b'_1 = b_1] \\
&= \frac{1}{2} \cdot (\Pr[b_2^{\mathcal{A}} = 0 | b_2 = 0, b'_1 = b_1] \cdot \Pr[b_2 = 0] + \\
&\qquad \Pr[b_2^{\mathcal{A}} = 1 | b_2 = 1, b'_1 = b_1] \cdot \Pr[b_2 = 1]) \\
&= \frac{1}{2} \cdot \left(p_0 \cdot \frac{1}{2} + p_1 \cdot \frac{1}{2}\right) \qquad \text{(Conditioned on } b'_1 = b_1) \\
&= \frac{1}{4} \cdot (p_0 + p_1)\,. \qquad (4)
\end{aligned}$$

We use $\overline{b_1}$ and $\overline{b_2}$ as shorthand for the additive inverse of b_1 mod 2 (i.e., $1 - b_1$) and for b_2, respectively.

Similarly, for case $(b_1, \overline{b_2})$,

$$\begin{aligned}\Pr[\mathbf{Exp}^{\epsilon\text{-ind-fa-ocpa}}_{\mathcal{B},\Pi}(\kappa, b_1, \overline{b_2}) = (b_1, b_2)] \\ &= \Pr[\widehat{b_1} = b_1] \cdot \Pr[b_2^{\mathcal{A}} \neq b_2] \\ &= \frac{1}{4} \cdot \left(\overline{p_0} + \overline{p_1}\right). \end{aligned} \tag{5}$$

To break the ϵ-IND-FA-OCPA notion, it suffices to show that its DP guarantee no longer holds for cases (b_1, b_2), $(b_1, \overline{b_2})$. We need to compare $\frac{1}{4} \cdot (p_0 + p_1)$ (Equation 4) and $\frac{1}{4} \cdot \left(\overline{p_0} + \overline{p_1}\right)$ (Equation (5)). We have the following inequality,

$$p_0 + p_1 > e^{t\epsilon} \cdot \overline{p_1} + e^{t\epsilon} \cdot \overline{p_0} + \mathsf{negl}(\kappa) + \mathsf{negl}(\kappa)$$

Apply Inequalities (2 and (3)), and we obtain

$$p_0 + p_1 > e^{t\epsilon} \cdot (\overline{p_1} + \overline{p_0}) + \mathsf{negl}(\kappa),$$

where $\mathsf{negl}(\kappa)$ is a negligible function. Then, multiply both the LHS and RHS by $\frac{1}{4}$, and substitute the notations for $p_0, p_1, \overline{p_0}, \overline{p_1}$. We finally obtain the following inequality,

$$\Pr[\mathbf{Exp}^{\epsilon\text{-ind-fa-ocpa}}_{\mathcal{B},\Pi}(\kappa, b_1, b_2) = (b_1, b_2)] > e^{t\epsilon} \cdot \Pr[\mathbf{Exp}^{\epsilon\text{-ind-fa-ocpa}}_{\mathcal{B},\Pi}(\kappa, b_1, \overline{b_2}) = (b_1, b_2)] + \mathsf{negl}(\kappa).$$

This means ϵ-IND-FA-OCPA no longer holds. The adversary $\mathcal{B}$ is efficient as it makes only one call to the PPT adversary $\mathcal{A}$ breaking the non-adaptive t-adjacent DP-Access experiment and consumes essentially the same resource as $\mathcal{A}$. This concludes the proof. □

C.3 Proof for Adaptive DP-Access of DO-Range

The proof for DP-Access of DO-Range is relatively straightforward. Before describing our proof, we quickly recap the necessary concepts in DO-Range and their notion of adaptive differential obliviousness [12], which we refer to as DO.

Overview. First, let $\mathbb{DS}$ denote a data structure, modeled as a stateful algorithm. This algorithm takes as input a security parameter and an operation, and outputs some addresses. These addresses denote the physical memory locations accessed by the operation and also include the operation type: *read* or *write*. The data structure $\mathbb{DS}$ can be either encrypted or unencrypted. However, it is irrelevant to DO since the adversary is only given the addresses, not $\mathbb{DS}$ itself. This distinction separates DP-Access from DO. A natural extension of DO in a private setting is to allow the adversary to choose the data structure and also be given the encrypted data structure.

In addition, there is a major difference between DO and DP-Access in defining the neighboring condition. In the DO definition (Definition 9), the data structure $\mathbb{DS}$ is fixed at the start. Then, to prepare the Left-or-Right challenge for differential privacy, *neighboring* in the data structure $\mathbb{DS}$ only comes from insertion queries. These insertion queries are as part of the N operations chosen adaptively by the adversary. They propose a *query-consistent neighbor-respecting* property to formally define this query restriction. Two sequences of operations (i.e., queries) are considered *neighboring* if and only if they are operations (including the data) of equal length, and only one operation differs. These two differing operations must be insertion operations. As a result, the underlying data structures from Left and Right are also neighboring after the operations. We will show later that this neighboring definition is equivalent to our neighboring condition in DP-Access with a certain instantiation of the abstract distance function.

Definition 9 (Adaptive Differential Obliviousness (DO) [11])**.** We say that a data structure $\mathbb{DS}$ satisfies adaptive (ϵ, δ)-differential obliviousness iff for any (possibly unbounded) stateful algorithm $\mathcal{A}$ that is query-consistent neighbor-respecting, for any N, $\mathcal{A}$'s view in the following two experiments $\mathbf{Exp}^{\mathrm{do},0}_{\mathcal{A}}(\lambda, N)$ and $\mathbf{Exp}^{\mathrm{do},1}_{\mathcal{A}}(\lambda, N)$ satisfy the following inequality:[21]

$$\Pr[\mathbf{Exp}^{\mathrm{do},0}_{\mathcal{A}}(\lambda, N) = 1] \leq e^{\epsilon(\lambda)} \cdot \Pr[\mathbf{Exp}^{\mathrm{do},1}_{\mathcal{A}}(\lambda, N) = 1] + \delta(\lambda).$$

$\mathbf{Exp}^{\mathrm{do},b}_{\mathcal{A}}(\lambda, N)$:

1: $\mathsf{addresses}_0 = \perp$
2: **for** $i = 1, 2, \ldots, N$ **do**
3: $\quad(\mathsf{op}_i^0, \mathsf{op}_i^1) \leftarrow \mathcal{A}(N, \mathsf{addresses}_{i-1})$
4: $\quad\mathsf{addresses}_i \leftarrow \mathbb{DS}(\lambda, \mathsf{op}_i^b)$
5: **end for**
6: $b' \leftarrow \mathcal{A}$
7: **return** b'

Figure 9: Adaptive Differential Obliviousness (DO) Experiment.

Theorem C.2. *If* DO-Range *satisfies adaptive differential obliviousness, then it also satisfies our adaptive* DP-Access *notion.*

Proof. To prove the theorem, we follow the standard reduction strategy, showing that if there exists a PPT adversary $\mathcal{A}$ that breaks adaptive DP-Access for DO-Range, then we can construct a PPT adversary $\mathcal{B}$ using $\mathcal{A}$ as a subroutine to break DO for DO-Range. We assume that we are working with some label space $\mathbb{L}$ and value space $\mathbb{V}$.

In this proof, we need to handle the following subtleties: translating the neighboring condition between the two definitions, DO and DP-Access, and addressing the support for insertion queries in DO that is not present in DP-Access.

First, we instantiate the following neighboring condition in DP-Access: let $\mathtt{dist}(\mathsf{ds}_0, \mathsf{ds}_1) = 1$ if and only if ds_0 and ds_1 differ in exactly one pair of label and value.

We construct $\mathcal{B}$ as follows. First, we initialize $\mathbb{DS}$ as an empty data structure over $\mathbb{L} \times \mathbb{V}$, denoted by $\mathbb{DS} \leftarrow \{\}$. This is not a special case for the DO definition, as the only way to populate the $\mathbb{DS}$ in DO is through insertions. In other words, there is no dedicated setup algorithm for DO experiment.

Adversary $\mathcal{B}$ can query any query-consistent neighbor-respecting operations up to some N. Suppose $\mathcal{B}$ makes t queries for $t \leq N$: $(\mathsf{op}_i^0, \mathsf{op}_i^1)$, for all $i \in [t]$, such that they satisfy the query-consistent neighbor-respecting property. Following the procedures of the DO experiment, given the random challenge bit b, for all $i \in [t]$, $\mathsf{addresses}_i \leftarrow \mathbb{DS}(\lambda, \mathsf{op}_i^b)$.

Before running $\mathcal{A}$ as a subroutine, for preparation, adversary $\mathcal{B}$ initializes empty data structures $\mathbb{DS}^0 = \mathbb{DS}^1 = \{\}$. Then $\mathcal{B}$

[21] From the bit guessing perspective, choosing this inequality may seem unusual. However, other inequalities are also ensured by the symmetric property of DP. In our DP-Access notion, we provide multiple inequalities.

runs $\mathbb{DS}^0(\lambda, \mathrm{op}_i^0)$ and $\mathbb{DS}^1(\lambda, \mathrm{op}_i^1)$ for all $i \in [t]$. Note that data structures $\mathbb{DS}^0$ and $\mathbb{DS}^1$ are stateful, so both $\mathbb{DS}^0$ and $\mathbb{DS}^1$ have been updated. They are neighboring data structures since $(\mathrm{op}_i^0, \mathrm{op}_i^1)$ for all $i \in [t]$ are query-consistent neighbor-respecting.

After that, $\mathcal{B}$ runs $\mathcal{A}$ internally that breaks the adaptive DP-Access security. The adaptive DP-Access experiment is given as input $(1^\lambda, \perp, \mathbb{DS}^0, \mathbb{DS}^1)$. The challenge bit of DP-Access is the same as the challenge bit of DO. The way $\mathcal{B}$ simulates the output eds_1 is as follows. Given the $(\mathrm{addresses}_i)_{i\in[t]}$ from running t operations on the $\mathbb{DS}$, for each insertion operation op_j^b, indicated by the write access in $\mathrm{addresses}_j$. $\mathcal{B}$ encrypts a memory block of 0's using the same IND-CPA blockcipher-based encryption scheme as in DO-Range, and attaches that to $\mathrm{addresses}_j$. For each read access indicated by the addresses, also attach the associated encrypted memory block. Then this means that $\mathcal{B}$ can simulate eds_1 as in DP-Access by returning all $(\mathrm{addresses}_i)_{i\in[t]}$, and associated encrypted memory blocks.

After the setup stage, $\mathcal{A}$ is given $\mathbb{DS}^0, \mathbb{DS}^1, \lambda, z = \perp, (\mathrm{addresses}_i)_{i\in[t]}$ and their associated encrypted memory blocks. We show how $\mathcal{B}$ responds to $\mathcal{A}$'s request. On $\mathcal{A}$'s adaptive find queries $\mathbf{q} = (q_i)_{i\in[s]}$, let $\mathcal{B}$ issue the same queries (q_i^0, q_i^1) for all $i \in [s]$ in the DO experiment; $\mathrm{addresses}'_i \leftarrow \mathbb{DS}(\lambda, q_i^b)$ for each $i \in [s]$. $\mathcal{B}$ returns $(\mathrm{addresses}'_i)_{i\in[s]}$ and the associated encrypted memory blocks as the query transcript. The eds_1 (from setup) is also returned to the $\mathcal{A}$ for each query, but this does not help much since eds_1 does not change in DO-Range from Find queries. At the end of the adaptive DP-Access experiment, $\mathcal{A}$ outputs a bit $b'_{\mathcal{A}}$. We let $\mathcal{B}$ output the same bit $b'_{\mathcal{A}}$ in the DO experiment.

This attack works straightforwardly as this is a game by rewriting. Let us consider the exact probabilities of $\mathcal{B}$. We choose the same $\epsilon(\lambda)$, and $\delta(\lambda)$, and pick the bits $b = 0$ and $b' = 1$ in the adaptive DP-Access notion, to be consistent with DO. Then for $\mathcal{A}$ breaking adaptive DP-Access, we have the following inequality,

$$\Pr\left[\mathbf{Exp}^{\text{dp-access},0}_{\mathcal{B},\mathcal{A},s}(1^\lambda, \perp, \mathbb{DS}^0, \mathbb{DS}^1) = 1\right] > e^{\epsilon(\lambda)} \cdot \Pr\left[\mathbf{Exp}^{\text{dp-access},1}_{\mathcal{B},\mathcal{A},s}(1^\lambda, \perp, \mathbb{DS}^0, \mathbb{DS}^1) = 1\right] + \delta(\lambda).$$

Let $s = N - t$, and substitute with the following equations,

$$\Pr\left[\mathbf{Exp}^{\text{dp-access},0}_{\mathcal{B},\mathcal{A},s}(1^\lambda, \perp, \mathbb{DS}^0, \mathbb{DS}^1) = 1\right] = \Pr\left[\mathbf{Exp}^{\text{do},0}_{\mathcal{B}}(\lambda, N) = 1\right],$$

and

$$\Pr\left[\mathbf{Exp}^{\text{dp-access},1}_{\mathcal{B},\mathcal{A},s}(1^\lambda, \perp, \mathbb{DS}^0, \mathbb{DS}^1) = 1\right] = \Pr\left[\mathbf{Exp}^{\text{do},1}_{\mathcal{B}}(\lambda, N) = 1\right].$$

Then we have the following inequality,

$$\Pr\left[\mathbf{Exp}^{\text{do},0}_{\mathcal{B}}(\lambda, N) = 1\right] > e^{\epsilon(\lambda)} \cdot \Pr\left[\mathbf{Exp}^{\text{do},1}_{\mathcal{B}}(\lambda, N) = 1\right] + \delta(\lambda).$$

This means the security guarantee of DO no longer holds. The adversary $\mathcal{B}$ is efficient as it runs internal PPT adversary $\mathcal{A}$ once that breaks the adaptive DP-Access security and consumes essentially the same resource as $\mathcal{A}$. This concludes the proof. □

D Experimental Evaluation (Extended)

Hardware. For convenience, we ran our experiments on a 24-Core AMD EPYC 7443P server with 315 GB of memory. We note that all our code is heavily single-threaded and can run on a standard laptop with 16 – 32 GB of memory in about one to two hours for the larger datasets.

Selecting the Bucket Size. For completeness, we provide more details of the parameter calculation. We based our calculations on the main theorem (cf. Theorem 6.3 [11]) that states the parameters of DO-Range for their adaptive differential obliviousness notion. Specifically, with $\epsilon = 4\epsilon_0 \log N$ and a negligible δ in the security parameter, which depends on the size of $\mathbb{L}$. For convenience, we let $\epsilon_0 = 1$ and $\log N = 18$. The main theorem imposes a constraint on the security parameter λ, which is $\log^*|\mathbb{L}| \leq \log\log\lambda$. Since $|\mathbb{L}| \in (16, 65536]$ is sufficient for our experiments, this implies $\log^*|\mathbb{L}| = 4$. It follows that $\lambda \geq 2^{16}$. The bucket size $Z = \frac{1}{\epsilon_0}\log^8\lambda$ is at least $2^{32} \approx 4.29 \times 10^9$ (cf. Section 5.5 [11]).

In the following, we show additional reconstructions on datasets for varying parameters.

D.1 Results for DO-Range

D.2 Results for ϵ-OPE-Range

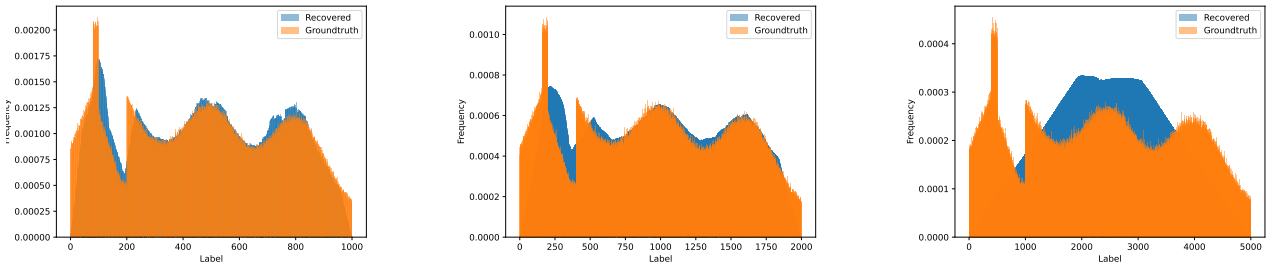
D.3 Datasets

IMDB Dataset. We use the "Actors IMDB" dataset to construct a dataset for age. The original dataset comprises 12, 915, 563 records of actors,[22] and it includes attributes like name, birth-year, (optional) death-year, and the titles they are known for. We construct a derived dataset of explicitly deceased actors, where the label value is their age of death. This yields a dataset of 207, 322 records.

Atlanta Dataset. The "Atlanta Employee Salary" dataset comprises 8246 records of government employees of the city of Atlanta for the year 2015.[23] For our experiments, we created two distinct databases where the primary label is either the age or salary attribute. This allows for real-world queries like "Find all employees of a certain age" or "Find all employees that have a salary falling into a range". For salary information, we pre-processed the data by rounding to the nearest 1K USD, resulting in salary labels ranging from 20K USD to 250K USD.

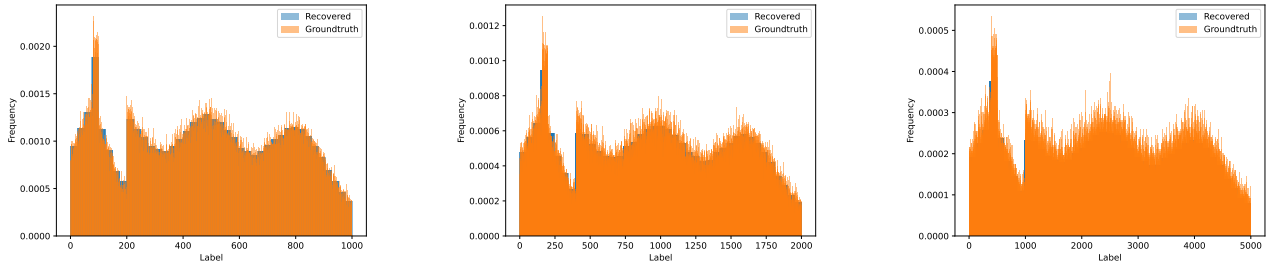
Synthetic Dataset. The synthetic datasets share the same underlying distribution and differ only in the number of records and the size of the label space. We provide these datasets with 2 million and 200,000 records. For the label space sizes we use $\{[0, 1000], [0, 2000], [0, 5000]\}$. The distribution is a mixture of normal, uniform, exponential, and Poisson distributions with varying parameters, as depicted in the groundtruth in Figure 11.

[22] https://developer.imdb.com/non-commercial-datasets/; downloaded on Oct 10, 2023.
[23] https://www.muckrock.com/foi/atlanta-325/city-employee-salary-information-atlanta-20915/



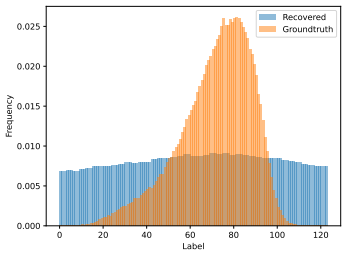
(a) Key range [0, 1000]. **(b) Key range [0, 2000].** **(c) Key range [0, 5000].**

Figure 10: DO-Range **with bucket size 512 on a synthetic dataset of 2 million records, sampled from key ranges (i.e., label ranges) [0, 1000], [0, 2000], and [0, 5000] following the same distribution. These ranges were chosen to the performance of our attack on larger key ranges. For ranges up to 2000 keys, our recovery yields good results. However, for larger key ranges, such as** $[0, 5000]$**, the recovery is less effective, requiring sufficient records. Each figure shows the recovery result for level 11 of** DO-Range **construction, where 80% of the keys were queried.**

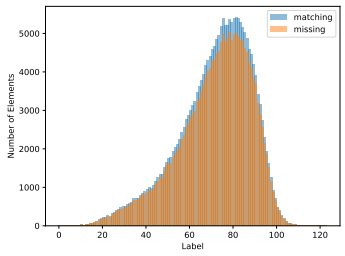


(a) Key range [0, 1000]. **(b) Key range [0, 2000].** **(c) Key range [0, 5000].**

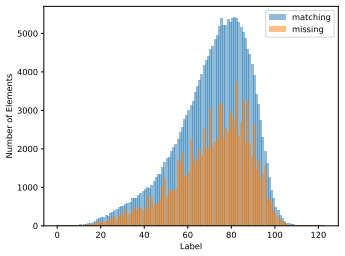
Figure 11: ϵ-OPE-Range **with** $\epsilon = 0.5$ **and** 80% **queried labels on a synthetic dataset of** $200,000$ **records, sampled from key ranges (i.e., label ranges) [0, 1000], [0, 2000], and [0, 5000] following the same distribution.**



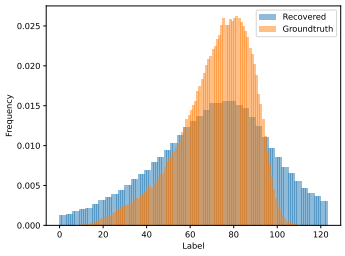
(a) Recovery for $\epsilon = 0.01$.



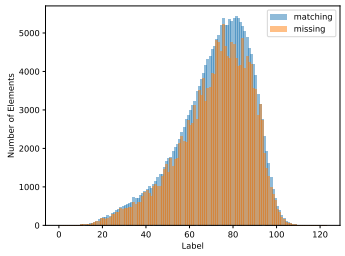
(b) Missing records for $\epsilon = 0.01$ when returning two additional partitions (as suggested by ϵ-OPE-Range).



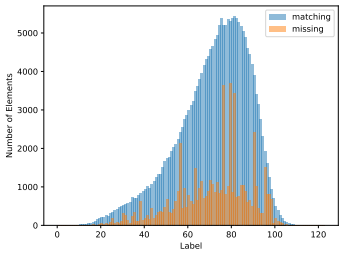
(c) Missing records for $\epsilon = 0.01$ when returning 20 additional partitions (this is half the key range of IMDB Age!).



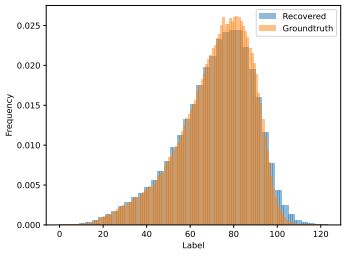
(d) Recovery for $\epsilon = 0.1$.



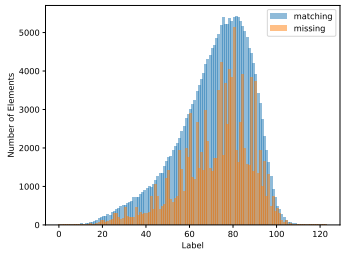
(e) Missing records for $\epsilon = 0.1$ when returning two additional partitions (as suggested by ϵ-OPE-Range).



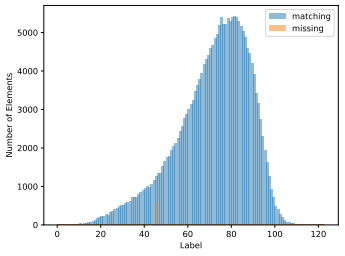
(f) Missing records for $\epsilon = 0.1$ when returning 20 additional partitions (this is half of the key range of IMDB Age!).



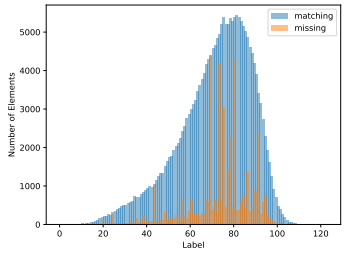
(g) Recovery for $\epsilon = 0.5$.



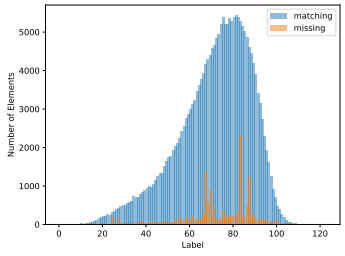
(h) Missing records for $\epsilon = 0.5$ when returning two additional partitions (as suggested by ϵ-OPE-Range).



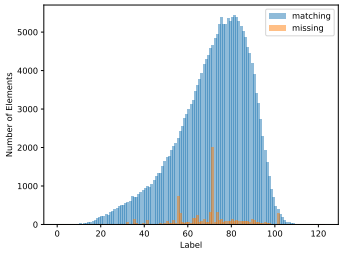
(i) Missing records for $\epsilon = 0.5$ when returning 20 additional partitions (this is half of the key range of IMDB Age!).



(j) Missing records for $\epsilon = 0.5$ when returning 6 additional partitions.

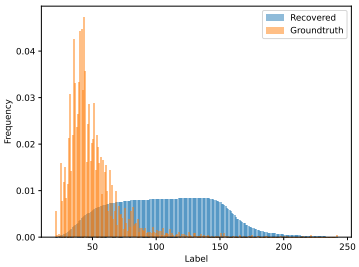


(k) Missing records for $\epsilon = 0.5$ when returning 8 additional partitions.

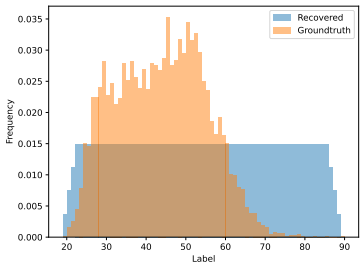


(l) Missing records for $\epsilon = 0.5$ when returning 10 additional partitions.

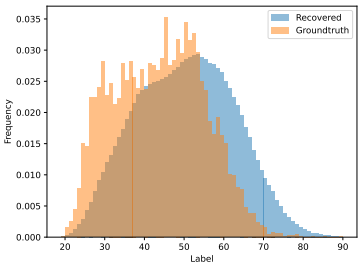
Figure 12: Recovery and utility results for ϵ-OPE-Range with $\epsilon = 0.01, 0.1, 0.5$, respectively, and 41 partitions on the IMDB Age dataset. We show the utility, i.e. how many records are missing from the response.



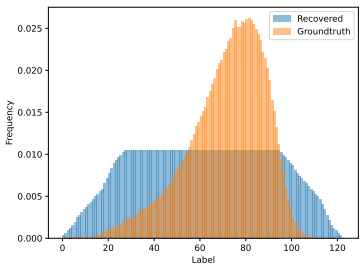
(a) Atlanta Salary; 100% labels queried; Level 9; reconstruction is inaccurate as the dataset contains too few records.



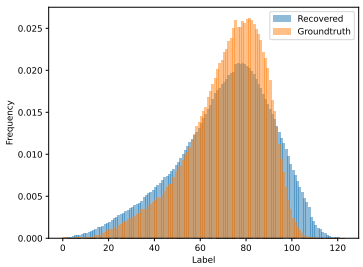
(b) Atlanta Age; 100% labels queried; Level 1



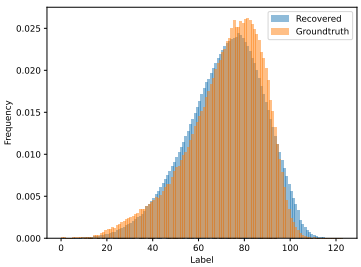
(c) Atlanta Age; 100% **labels queried; Level** 9



(d) IMDB Age; 70% labels queried; Level 4

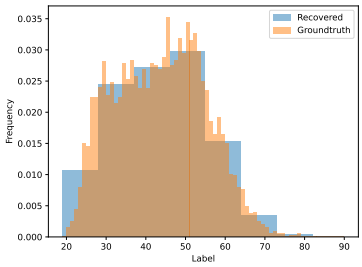


(e) IMDB Age; 70% labels queried; Level 9

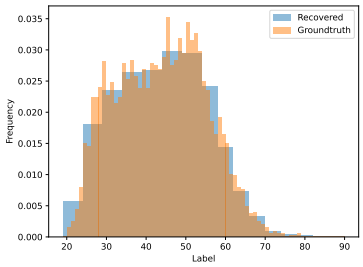


(f) IMDB Age; 70% labels queried; Level 13

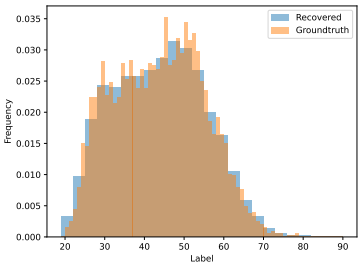
Figure 13: Extra Figures for DO-Range**.**



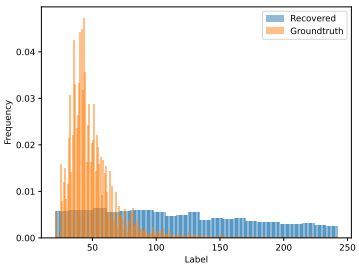
(a) Atlanta Age; $\epsilon = 1$**;** 8 **partitions**



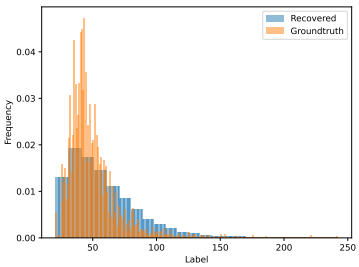
(b) Atlanta Age; $\epsilon = 1$**;** 16 **partitions**



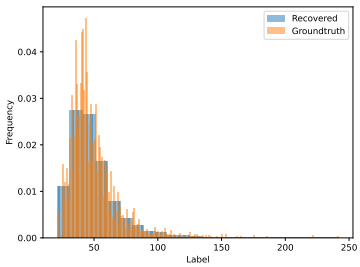
(c) Atlanta Age; $\epsilon = 1$**;** 24 **partitions**



(d) Atlanta Salary; $\epsilon = 0.01$**;** 24 **partitions**



(e) Atlanta Salary; $\epsilon = 0.1$**;** 24 **partitions**



(f) Atlanta Salary; $\epsilon = 0.5$**;** 24 **partitions**

Figure 14: Recovery for ϵ-OPE-Range **on the Atlanta Salary and Age dataset with** 70% **of the labels queried.**