ABSTRACT

Ciphertexts of an order-preserving encryption (OPE) scheme preserve the order of their corresponding plaintexts. However, OPEs are vulnerable to inference attacks that exploit this preserved order. Differential privacy (DP) has become the de-facto standard for data privacy. One of the most attractive properties of DP is that any post-processing computation, such as inference attacks, performed on the noisy output of a DP algorithm does not degrade its privacy guarantee. In this work, we propose a novel \textit{differentially private order preserving encryption} scheme, \textit{OPE}. Under \textit{OPE}, the leakage of order from the ciphertexts is differentially private. Consequently, in the least, \textit{OPE} ensures a formal guarantee (a relaxed DP guarantee) even in the face of inference attacks. To the best of our knowledge, this is the first work to combine DP with a OPE. \textit{OPE} is based on a novel \textit{differentially private order preserving encoding} scheme, \textit{OPE}, that can be of independent interest in the local DP setting. We demonstrate \textit{OPE}'s utility in answering range queries via empirical evaluation on four real-world datasets. For instance, \textit{OPE} misses only around 4 in every 10K correct records on average for a dataset of size $\sim 732K$ with an attribute of domain size $\sim 18K$ and $\epsilon = 1$.

CCS CONCEPTS

- Security and privacy → Cryptography; Management and querying of encrypted data.

KEYWORDS

Inference Attack; Encrypted Database; Range Query

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

ACM Reference Format:
Amrita Roy Chowdhury, Bolin Ding, Somesh Jha, Weiran Liu, and Jingren Zhou. 2022. Strengthening Order Preserving Encryption with Differential Privacy. In Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security (CCS ’22), November 7–11, 2022, Los Angeles, CA, USA. ACM, New York, NY, USA, 23 pages. https://doi.org/10.1145/3548606.3560610

1 INTRODUCTION

Frequent mass data breaches [7, 11–15] of sensitive information have exposed the privacy vulnerability of data storage in practice. This has led to a rapid development of systems that aim to protect the data while enabling statistical analysis on the dataset, both in academia [27, 38, 66, 77, 100] and industry [8–10, 71, 105]. Encrypted database systems that allow query computation over the encrypted data is a popular approach in this regard. Typically, such systems rely on \textit{property-preserving encryption} schemes [30, 35] to enable efficient computation. \textit{Order-preserving encryption} (OPE) [23, 35, 78, 92, 98] is one such cryptographic primitive that preserves the numerical order of the plaintexts even after encryption. This allows actions like sorting, ranking, and answering range queries to be performed directly over the encrypted data [23, 56, 68, 75, 87, 89, 90].

However, encrypted databases are vulnerable to inference attacks [32, 51, 61, 62, 65, 80–82, 85, 86, 97] that can reveal the plaintexts with good accuracy. Most of these attacks are inherent to any property-preserving encryption scheme – they do not leverage any weakness in the cryptographic security guarantee of the schemes but rather exploit just the \textit{preserved property}. For example, the strongest cryptographic guarantee for OPEs (IND-FA-OPCA, see Sec. 2.2) informally states that only the order of the plaintexts will be revealed from the ciphertexts. However, inference attacks [61, 62, 65] can be carried out by leveraging only this ordering information. The basic principle of these attacks is to use auxiliary information to estimate the plaintext distribution and then correlate it with the ciphertexts based on the preserved property [55].

Differential privacy (DP) has emerged as the de-facto standard for data privacy with widespread adoption in practice [17, 49, 53, 54, 58, 73, 91, 109]. DP is an information theoretic guarantee that provides a rigorous guarantee of privacy for individuals in a dataset regardless of an adversary’s auxiliary knowledge [107]. It is characterized by a parameter $\epsilon > 0$ where lower the value of $\epsilon$, greater the privacy guarantee achieved. An additional appealing property of DP is that any post-processing computation, such as inference attacks, performed on the noisy output of a DP algorithm does not incur additional privacy loss.

In this work, we ask the following question:

\textbf{Is it possible to leverage the properties of DP for providing a formal security guarantee for OPEs even in the face of inference attacks?}

To this end, we propose a novel \textit{differentially private order preserving encryption} scheme, \textit{OPE}. Recall that standard OPE schemes are designed to reveal nothing but the order of the plaintexts. Our proposed scheme, \textit{OPE}, ensures that this leakage of order is differentially private. In other words, the cryptographic guarantee of OPEs is strengthened with a layer of DP guarantee (specifically, a relaxed definition of DP as discussed in the following paragraph). As a result, even if the cryptographic security guarantee of standard OPEs proves to be inadequate (in the face of inference attacks), the
DP guarantee would continue to hold true. Intuitively, the reason is that DP is resilient to post-processing computations as discussed above. To the best of our knowledge, this is the first work to combine DP with a property-preserving encryption scheme.

1.1 Brief Overview of Key Ideas
The standard DP guarantee requires any two pairs of input data to be indistinguishable from each other (see Sec. 2.1) and is generally catered towards answering statistical queries over the entire dataset. However, in our setting we require the output of the DP mechanism to retain some of the ordinal characteristics of its input – the standard DP guarantee is not directly applicable to this case. Hence, we opt for a natural relaxation of DP – only pairs of data points that are ‘close’ to each other should be indistinguishable. Specifically, the privacy guarantee is heterogeneous and degrades linearly with the $\ell_1$-distance between a pair of data points. It is denoted by $\epsilon$-dLDP (or $\epsilon$-dDP in the central model of DP; see Sec. 2.1). This relaxation is along the lines of $d_k$-privacy [41] and is amenable to many practical settings. For instance, consider a dataset of annual sale figures of clothing firms. The information whether a firm is a top seller or a mid-range one is less sensitive than its actual sales figures. Similarly, for an age dataset, whether a person is young or middle-aged is less sensitive than their actual age.

DP guarantee inherently requires randomization – this entails an inevitable loss of utility, i.e., some pairs of output might not preserve the correct order of their respective inputs. In order to reduce the instances of such pairs, OPe offers the flexibility of preserving only a partial order of the plaintexts. Specifically, a (user specified) partition is defined on the input domain and the preserved order is expected at the granularity of this partition. The output domain is defined by a numeric encoding over the intervals of the partition and all the elements belonging to the same interval are mapped to the corresponding encoding for the interval (with high probability). Due to the linear dependence of the DP guarantee (and consequently, the ratio of output probabilities) on the distance between the pair of inputs, lower is the number of intervals in the partition, higher is the probability of outputting the correct encoding in general (see Sec. 3.2 and Sec. 7.2). OPe preserves the order over this encoding. The reason why this results in better utility for encrypted databases is illustrated by the following example. The typical use case for OPe encrypted databases is retrieving a set of records from the outsourced database that belong to a queried range. Suppose a querier asks for a range query $[a, b]$ and let $P$ be a partition that covers the range with $k$ intervals $\{[s_1, e_1], \ldots, [s_k, e_k]\}$ such that $s_1 < a < e_1$ and $s_k < b < e_k$. A database system encrypted with OPe and instantiated with the partition $\mathcal{P}$ will return all the records that are noisily mapped to the range $[s_1, e_2]$ (since the order is preserved at the granularity of $\mathcal{P}$). Thus, the querier has to pay a processing overhead of fetching extra records, i.e., the records belonging to the ranges $\{[s_1, a-1], [b+1, e_k]\}$. However, if $k < b-a$, then with high probability it would receive all the correct records in $[a, b]$ which can be decrypted and verified (Sec. 6). To this end, we first propose a new primitive, OPec, that enables order preserving encoding with $\epsilon$-dLDP. The encryption scheme, OPec, is then constructed using the OPe primitive and a OPE (Sec. 4).

Our work is along the lines of a growing area of research exploring the association between DP and cryptography [28, 45, 76, 104] (see Sec. 8). Beyond OPEs, the OPec primitive can be used as a building block for other secure computation that require ordering, such as order-revealing encryptions (see full paper [46]). Additionally, OPec can be of independent interest for the LDP setting in answering a variety of queries, such as ordinal queries, frequency and mean estimation (see Sec. 3.3).

1.2 Discussion
In this section, we answer some key questions pertinent to our work that the readers might have.

Q1. Why should we care about OPEs?
- A. Range query constitutes an extremely important class of queries for data analytics. For instance, about half of the queries of the TPC-H benchmark [1], which is the standard benchmark for OLAP queries, have range predicates [50]. Additionally, range query is a fundamental operation in DBMS with its applications ranging from B-tree indices [29, 72] to band-joins [48]. Thus, efficient support for range queries on encrypted databases is a fundamental task for secure data analytics. One of the main challenges of practical deployment of cryptographic protocols is the associated performance overhead especially with large realistic datasets. The advantage of OPEs in this regard is that it allows range queries to be performed directly over the encrypted data thereby matching the optimal performance of plaintext computation. Hence, given this immense performance advantage, OPEs are a key building block for encrypted databases [55] and exploring secure implementations of OPEs is still an important problem.

Q2. What is the advantage of a OPE scheme over just OPec primitive or a OPE scheme?
- A. OPe satisfies a new security guarantee, $\epsilon$-IND-FA-OCPA, (see Sec. 4.2) that enhances the cryptographic guarantee of a OPE scheme (IND-FA-OCPA) with a layer of $\epsilon$-dDP guarantee. As a result, OPe enjoys strictly stronger security than both OPec primitive ($\epsilon$-dDP) and OPE (IND-FA-OCPA).

Q3. What are the security implications of OPe in the face of inference attacks?
- A. In the very least, OPe rigorously limits the accuracy of inference attacks for every record for all adversaries (Thm. 7, Sec. 5). In other words, OPe guarantees that none of the attacks can infer the value of any record beyond a certain accuracy that is allowed by the dLDP guarantee. For instance, for an age dataset and an adversary with real-world auxiliary knowledge, no inference attack in the snapshot model can distinguish between two age values $(x, x')$ such that $|x-x'| \leq 8$ for $\epsilon = 0.1$ (Sec. 7.2).

Q4. How is OPe’s utility (accuracy of range queries)?
- A. We present a construction for the OPec primitive (and subsequently, OPe) and our experimental results on four real-world datasets demonstrate its practicality for real-world usage (Sec. 7). Specifically, OPe misses only 4 in every 10K correct records on average for a dataset of size $\sim$ 732K with an attribute of domain size 18K and $\epsilon = 1$. The overhead of processing extra records is also low – the average number of extra records returned is just 0.3% of the total dataset size.

Q5. When to use OPe?
- A. As discussed above, OPe gives a strictly stronger guarantee than
any OPE scheme (even in the face of inference attacks) with almost no extra performance overhead (Sec. 6). Additionally, it is backward compatible with any encrypted database that is already using a OPE scheme (satisfying IND-FA-OC PA, see Sec. 6). Hence, OPe could be used for secure data analytics in settings where (1) the e-dDP guarantee is acceptable, i.e., the main security concern is preventing the distinction between input values close to each other (such as the examples discussed above) and (2) the application can tolerate a small loss in utility. Specifically in such settings, replacing encrypted databases with OPe would give a strictly stronger security guarantee against all attacks with nominal change in infrastructure or performance—a win-win situation.

2 BACKGROUND

2.1 Differential Privacy

Differential privacy is a quantifiable measure of the stability of the output of a randomized mechanism to changes to its input. There are two popular models of differential privacy, local and central. The local model consists of a set of individual data owners and an untrusted data aggregator; each individual perturbs their data using a local DP algorithm and sends it to the aggregator which uses these noisy data to infer some statistics on the entire dataset. Thus, the LDP model allows gleaning of useful information from the dataset without requiring the data owners to trust any third-party entity. The LDP guarantee is formally defined as follows:

Definition 2.1 (Local Differential Privacy, LDP). A randomized algorithm \( M : \mathcal{X} \rightarrow \mathcal{Y} \) is \( \epsilon \)-LDP if for any pair of private values \( x, x' \in \mathcal{X} \) and any subset of output \( T \subseteq \mathcal{Y} \)

\[
\Pr[M(x) \in T] \leq e^{\epsilon} \cdot \Pr[M(x') \in T]
\]

The above definition is equivalent to the notion of metric-based LDP [24, 41] where the metric used is \( \ell_1 \)-distance.

In the central differential privacy (CDP) model, a trusted data curator collates data from all the individuals and stores it in the clear in a centrally held dataset. The curator mediates upon every query posed by a mistrustful analyst and enforces privacy by adding noise to the answers of the analyst’s queries before releasing them.

Definition 2.2 (Distance-based Local Differential Privacy, dLDP). A randomized algorithm \( M : \mathcal{X} \rightarrow \mathcal{Y} \) is \( \epsilon \)-based locally differentially private (or \( \epsilon \)-dLDP), if for any pair of private values \( x, x' \in \mathcal{X} \) and any subset of output \( T \subseteq \mathcal{Y} \)

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

The notion of adjacent inputs is application-dependent, and typically means that \( X \) and \( X' \) differ in a single element (corresponding to a single individual). Particularly in our setting, the equivalent definition of the distance based relaxation of differential privacy in the CDP model is given as follows:

Definition 2.4 (Distance-based Central Differential Privacy, dDP). A randomized algorithm \( M : \mathcal{X} \rightarrow \mathcal{Y} \) is \( \epsilon \)-differential based centrally differentially private (or \( \epsilon \)-dDP), if for any pair of datasets \( X \) and \( X' \) such that they differ in a single element, \( x_i \) and \( x'_i \), and any subset of output \( T \subseteq \mathcal{Y} \),

\[
\Pr[M(X) \in T] \leq e^{\epsilon|x_i-x'_i|} \cdot \Pr[M(X') \in T]
\]

We define \( X \) and \( X' \), as described above, to be \( t \)-adjacent where \( t \geq |x_i-x'_i| \), i.e., the differing elements differ by at most \( t \). Trivially, any pair of \( t \)-adjacent datasets is also \( t' \)-adjacent for \( t' > t \). Next, we formalize the resilience of dLDP (and dDP) to post-processing computations.

Theorem 1 (Post-Processing [52]). Let \( M : \mathcal{X} \rightarrow \mathcal{Y} (M : \mathcal{X} \rightarrow \mathcal{Y}) \) be an \( \epsilon \)-dLDP (dDP) algorithm. Let \( g : \mathcal{Y} \rightarrow \mathcal{Y}' \) be any randomized mapping. Then \( g \circ M \) is also \( \epsilon \)-dLDP (dDP).

2.2 Order Preserving Encryption

In this section, we discuss the necessary definitions for OPEs.

Definition 2.5 (Order Preserving Encryption [92]). An order preserving encryption (OPE) scheme \( \mathcal{E} = (K, E, D) \) is a tuple of probabilistic polynomial time (PPT) algorithms:

- **Key Generation** \( (K) \). The key generation algorithm takes as input a security parameter \( \lambda \) and outputs a secret key (or state) \( S \) as \( S \leftarrow \mathcal{K}(\lambda) \).
- **Encryption** \( (E) \). Let \( X = \langle x_1, \cdots, x_n \rangle \) be an input dataset. The encryption algorithm takes as input a secret key \( S \), a plaintext \( x \in \mathcal{X} \), and an order \( \Gamma \) (any permutation of \( \{1, \cdots, n\} \) ). It outputs a new key \( S' \) and a ciphertext \( y \) as \( (S', y) \leftarrow \mathcal{E}(S, x, \Gamma) \).
- **Decryption** \( (D) \). Decryption recovers the plaintext \( x \) from the ciphertext \( y \) using the secret key \( S \), \( x \leftarrow \mathcal{D}(S, y) \).

Additionally, we have

- **Correctness Property**: \( x \leftarrow \mathcal{D}(\mathcal{E}(S, x, \Gamma)) \), \( S, \forall x \in \mathcal{X}, \forall \Gamma \)
- **Order Preserving Property**: \( x > x' \implies y > y', \forall x, x' \) where \( y (y') \) is the ciphertext corresponding to the plaintext \( x (x') \).

The role of the order \( \Gamma \) in the above definition is discussed later in this section. The strongest formal guarantee for a OPE scheme is indistinguishability against frequency-analyzing ordered chosen plaintext attacks (IND-FA-OC PA). We present two definitions in connection to this starting with the notion of randomized orders as defined by Kerschbaum [78].

Definition 2.6. (Randomized Order [78]) Let \( X = \langle x_1, \cdots, x_n \rangle \) be a dataset. An order \( \Gamma = \langle y_1, \cdots, y_n \rangle \), where \( y_i \in [n] \) and \( i \neq j \implies y_i \neq y_j \), for all \( i, j \) of dataset \( X \), is defined to be a random ordered if it holds that

\[
\forall i, j (x_i > x_j \implies y_i > y_j) \wedge (y_i > y_j \implies x_i \geq x_j)
\]

For a plaintext dataset \( X \) of size \( n \), a randomized order, \( \Gamma \), is a permutation of the plaintext indices \( \{1, \cdots, n\} \) such that its inverse, \( \Gamma^{-1} \), gives a sorted version of \( X \). This is best explained by an example—let \( X = (9, 40, 15, 76, 15, 76) \) be a dataset of size 6. A randomized

1See full paper [46] for additional notes.
order for $X$ can be either of $\Gamma_1 = (1, 4, 2, 5, 3, 6)$, $\Gamma_2 = (1, 4, 3, 5, 2, 6)$, $\Gamma_3 = (1, 4, 2, 6, 3, 5)$ and $\Gamma_4 = (1, 4, 3, 6, 2, 5)$. This is because the order of the two instances of 75 and 15 does not matter for a sorted version of $X$.

**Definition 2.7** (IND-FA-OCPA [78, 92]). An order-preserving encryption scheme $E = (K, E, D)$ has indistinguishable ciphertexts under frequency-analyzing ordered chosen plaintext attacks if for any PPT adversary $A_{\text{ppt}}$:

$$\Pr[G_{\text{FA-OCPA}}^E(\kappa, 1) = 1] - \Pr[G_{\text{FA-OCPA}}^E(\kappa, 0) = 1] \leq \negl(\kappa)$$

where $\kappa$ is a security parameter, $\negl(\cdot)$ denotes a negligible function and $G_{\text{FA-OCPA}}^E(\kappa, b)$ is the random variable denoting $A_{\text{ppt}}$’s output for the following game:

**Game $G_{\text{FA-OCPA}}^E(\kappa, b)$**

1. $(X_0, X_1) \leftarrow A_{\text{ppt}}$ where $|X_0| = |X_1| = n$ and $X_0$ and $X_1$ have at least one common randomized order.
2. Select $\Gamma'$ uniformly at random from the common randomized orders of $X_0, X_1$.
3. $S_0 \leftarrow (\Gamma^4, 1)$.
4. For $\forall i \in [n]$, run $(S_i, y_{b_i}) \leftarrow E(S_i, x_{b_i}, \Gamma')$.
5. Run $b' \leftarrow A_{\text{ppt}}(y_{b_1}, \cdots, y_{b_n})$ where $b'$ is $A_{\text{ppt}}$’s guess for $b$.

Informally, this guarantee implies that nothing other than the order of the plaintexts, not even the frequency, is revealed from the ciphertexts. Stated otherwise, the ciphertexts only leak a randomized order of the plaintexts (randomized orders do not contain any frequency information since each value always occurs exactly once) which is determined by the input order $\Gamma$ in Defn. 2.5. In fact, if $\Gamma$ itself happens to be a randomized order of the input $X$ then, the randomized order leaked by the corresponding ciphertexts is guaranteed to be $\Gamma$. For example, for $X = (9, 40, 15, 76, 15, 76)$ and $\Gamma = (1, 4, 2, 5, 3, 6)$, we have $y_i < y_3 < y_5 < y_2 < y_4 < y_6$ ($y_i$ denotes the corresponding ciphertext for $x_i$) and $\Gamma^{-1} = (1, 3, 5, 2, 4, 6)$. Thus, the IND-FA-OCPA guarantee ensures that two datasets with a common randomized order – but different plaintext frequencies – are indistinguishable. For example, in the aforementioned game $G_{\text{FA-OCPA}}^E(\cdot, A_{\text{ppt}})$ would fail to distinguish between the plaintext datasets $X_0 = (9, 40, 15, 76, 15, 76)$ and $X_1 = (22, 94, 23, 94, 36, 94)$, both of which share the randomized order $\Gamma^4 = (1, 4, 2, 5, 3, 6)$.

### 3 ε-dLDP ORDER PRESERVING ENCODING (OPec)

In this section, we discuss our proposed primitive – ε-dLDP order preserving encoding, OPec. First, we define the OPec primitive and its construction. Next, we describe how to use the OPec primitive to answer queries in the LDP setting.

**Notations.** $[n], n \in \mathbb{N}$ denotes the set $\{1, 2, \cdots, n - 1, n\}$. If $X = [s, e]$ is an input domain, then a $k$-partition $\mathcal{P}$ on $X$ denotes a set of $k$ non-overlapping intervals $X_i = [s_i, e_i]$, $s_{j+1} = e_j, j \in [k-1]$ such that $\bigcup_{i=1}^{k} X_i = X$. For example, for $X = [1, 100]$, $\mathcal{P} = \{(1, 10], (10, 20], \cdots, (90, 100]\}$ denotes a 10-partition. Let $c$

\[c\text{-}

The first interval, $X_1 = [s_1, e_1]$, is a closed interval.

\[\text{The domain of partitions defined over } X. \text{ Additionally, let } O = \{o_1, \cdots, o_k\}, o_i < o_{i+1}, i \in [k-1] \text{ represent the output domain where } o_i \text{ is the corresponding encoding for the interval } X_i \text{ and let } \mathcal{P}(x) = o_i \text{ denote that } x \in X_i. \text{ Referring back to our example, if } O = \{1, 2, \cdots, 10\}, \text{ then } \mathcal{P}(45) = 5.\]

#### 3.1 Definition of OPec

OPec is a randomised mechanism that encodes its input while maintaining some of its ordinality.

**Definition 3.1** (ε-dLDPOrder Preserving Encoding, OPec). For a given $k$-partition $\mathcal{P} \in X$, a $\varepsilon$-dLDP order preserving encoding scheme, $\text{OPec} : X \times X \times \mathbb{R}_{\geq 0} \mapsto O$ is a randomized mechanism such that

\[\Pr[\mathcal{P}(x, \mathcal{P}, \varepsilon) = o_i] > \Pr[\mathcal{P}(x, \mathcal{P}, \varepsilon) = o_j] \]  

\[\implies \exists o \in \mathcal{P} \text{ such that,} \]  

\[\Pr[O_{\text{opce}}(x, \mathcal{P}, \varepsilon) = o_i] > \Pr[O_{\text{opce}}(x, \mathcal{P}, \varepsilon) = o_j] \]

\[\Pr[O_{\text{opce}}(x, \mathcal{P}, \varepsilon) = o_i] \leq e^{\varepsilon|X-x|} \cdot \Pr[O_{\text{opce}}(x', \mathcal{P}, \varepsilon) = o_i] \]

The first property in the above definition signifies the flexibility of the OPec primitive to provide only a partial ordering guarantee. For instance, in our above example $k = 10 \leq |X| = 100$. Thus, $\mathcal{P}$ acts as a utility parameter – it determines the granularity at which the ordering information is maintained by the encoding (this is independent of the privacy-accuracy trade-off arising from the choice of $\varepsilon$). For example, for the same value of $\varepsilon$ and $X = [1, 100]$, $\mathcal{P} = \{[1, 10], (10, 20], \cdots, (90, 100]\}$ gives better utility than $\mathcal{P}' = \{(1, 33], (33, 66], (66, 100]\}$ since the former preserves the ordering information at a finer granularity. $\mathcal{P} = O = X$ denotes the default case where effectively no partition is defined on the input domain and $\mathcal{P}(x) = x, x \in X$ trivially. We discuss the significance of the parameter $\mathcal{P}$ in Sec. 6.

Due to randomization (required for the dLDP guarantee), OPec is bound to incur some errors in the resulting numerical ordering of its outputs. To this end, the second property guarantees that the noisy output is most likely to be either the correct one or the ones immediately next to it. For instance, for the aforementioned example, OPec(45, $\mathcal{P}$, $\varepsilon$) is most likely to fall in $\{4, 5, 6\}$. This ensures that the noisy outputs still retain sufficient ordinal characteristics of the corresponding inputs. Note that the actual value of the encodings in $O$ does not matter at all as long as the ordinal constraint $o_i < o_{i+1}, i \in [k - 1]$ is maintained. For instance for $\mathcal{P} = \{(1, 10], (10, 20], \cdots, (90, 100]\}, O = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}, \mathcal{O}' = \{5, 15, 25, 35, 45, 55, 65, 75, 85, 95\}$ and $\mathcal{O}'' = \{81, 99, 120, 150, 234, 345, 400, 432, 536, 637\}$ are all valid.

Finally, the third property ensures that the primitive satisfies $\varepsilon$-dLDP. Note that $\varepsilon = \infty$ represents the trivial case $O_{\text{opce}}(X, \mathcal{P}, \infty) = \mathcal{P}(X)$.

#### 3.2 Construction of OPec

In this section, we describe a construction for the OPec primitive (Alg. 1). The algorithm is divided into two stages. In Stage 1 (Steps 1-3), it computes the central tendency (a typical value for a distribution) $[102]$, $d_i, i \in [k]$, of each of the intervals of the given
Algorithm 1 Construction of OPeC

Setup Parameters: \( \mathcal{D} \) - Prior input distribution over \( X \), its default value is the uniform distribution; 
\( \mathcal{O} \) - Output domain \( \{o_1, \ldots, o_k\} \); 
Input: \( x \) - Number to be encoded via OPeC; \( \epsilon \) - Privacy budget; 
\( \mathcal{P} \) - A \( k \)-partition \( \{\{s_1, e_1\}, \ldots, \{s_k, e_k\}\} \) over \( X \) 
Output: \( o \) - Output encoding;

Stage I: Computation of central tendency for each interval
1. for \( i \in [k] \)
2. \( d_i = \text{Weighted median of the interval} (s_i, e_i) \) where \( \mathcal{D} \) gives the corresponding weights
3. end for

Stage II: Computation of the output probability distributions
4. for \( x \in X \) 
5. for \( i \in [k] \)
6. \[ p_{k,i} = \frac{e^{-|x-d_i|/\epsilon/2}}{\sum_{j=1}^{k} e^{-|x-d_j|/\epsilon/2}} \] \( \triangleright \) \( p_{k,i} = \Pr [\text{OPeC}(x, \mathcal{P}, \epsilon) = o_i] \) (7)
7. end for
8. \( p_x = \{p_{k,1}, \ldots, p_{k,k}\} \) \( \triangleright \) Encoding (output) probability distribution for \( x \)
9. end for
10. \( o = p_x \) \( \triangleright \) Encoding drawn at random from the distribution \( p_x \)
11. Return \( o \)

\( k \)-partition \( \mathcal{P} \). Specifically, we use weighted median [47] as our measure for the central tendency where the weights are determined by a prior on the input data distribution, \( \mathcal{D} \). This maximizes the expected number of inputs that are mapped to the correct encoding, i.e., \( x \) is mapped to \( \mathcal{P}(x) \). \( \mathcal{D} \) can be estimated from domain knowledge or (non-private) auxiliary datasets. In the event such a prior is not available, \( \mathcal{D} \) is assumed to be the uniform distribution \( (d_i \text{ is the median}) \).

In Stage II (Steps 4-9), the encoding probability distributions are computed such that the probability of \( x \) outputting the \( i \)-th encoding, \( o_i \), is inversely proportional to its distance from the \( i \)-th central tendency, \( d_i \). Specifically, we use a variant of the classic exponential mechanism [52, 67] (Eq. (7)). An illustration of the algorithm is in the full paper [46].

Theorem 2. Alg. 1 gives a construction for OPeC (Def. 3.1).

The proof of the above theorem follows directly from two facts. First, Alg. 1 satisfies the ordinal constraint of Eq. 6 (Lemma 11 in the full paper [46]) as depicted in Fig. 1. Second, it is straightforward from Eq. 7 that Alg. 1 satisfies \( \epsilon \)-dLDP (Lemma 12 in the full paper [46]).

Size of partition \( |\mathcal{P}| \). From Eq. 7, we observe that for every input \( x \), the encoding probability distribution \( p_x \) is an exponential distribution centered at \( \mathcal{P}(x) \) – its correct encoding. Moreover, the smaller is the size of \( \mathcal{P} \) (number of intervals in \( \mathcal{P} \)), the larger is the probability of outputting \( \mathcal{P}(x) \) (or its immediate neighbors). This is demonstrated in Fig. 1 which plots \( p_x \) for \( x = 50 \) and \( \epsilon = 0.1 \) under varying equi-length partitioning of the input domain [100].

Remark 1. The dLDP guarantee of OPeC (Thm. 2) does not depend on the partition \( \mathcal{P} \). Thus, the partition size could range from \( k = |\mathcal{O}| = |X| \) (no effective partitioning at all) to \( k = 2 \). Additionally, the dLDP guarantee (and utility) is also independent of the encoding domain, \( \mathcal{O} \), as long as the appropriate ordering constraint is valid. Also note that the partition can be completely arbitrary.

Design Choices. Note that we intend to use the OPeC primitive as a building block for our differentially private OPe scheme (details in Sec. 4). Now for a standard OPe, the utility (accuracy) remains exactly the same as that of the plaintext. However, the inherent randomization in DP results in an inevitable loss in utility. Hence, our primary motivation for designing OPeC is to (1) provide a meaningful guarantee against inference attacks (2) with high utility.

Why dLDP? The standard (local) DP definition requires every input pair to be indistinguishable – this requires the addition of a large amount of noise resulting in low utility, especially for large data domains. With dLDP, only input pairs that are close to each other are indistinguishable which still results in a meaningful guarantee in practice (Sec. 1.1). Essentially, this heterogeneous guarantee reveals some controlled information about the \( f_1 \)-distance between input pairs. Note that utility in our context implies how well is the order of the plaintexts preserved. Observe that the order of two values is determined by their \( f_1 \)-distance. Thus intuitively, the output under dLDP retains some ordinal information about its input, thereby improving utility.

Why partitioning? The typical approach to achieve DP is via addition of noise that is proportional to a domain-dependent term called sensitivity [52]. Common approaches to mitigate the cost of high sensitivity (for large data domains), such as propose-test-release [52], result in approximate DP. Partitioning bypasses the need for noise addition by design and provides a clean way to improve utility with pure DP (see Sec. 6 for details). Additionally, partitioning provides flexibility. A use case is demonstrated below where one can plug-and-play with different values of \( \mathcal{P} \) enabling OPeC to answer different types of queries in the LDP setting.

3.3 LDP Mechanisms using OPeC

The OPeC primitive can be of independent interest in the LDP setting. Depending on the choice of the partition \( \mathcal{P} \) over the input
domain $X$, ОРп can be used to answer different types of queries with high utility. In this section, we describe how to use ОРп to answer two such queries.

### Problem Setting
We assume the standard LDP setting with $n$ data owners, $DO_i, i \in [n]$ each with a private data $x_i$.

### Ordinal Queries
ОРп can be used to answer queries in the LDP setting that require the individual noisy outputs to retain some of the ordinal characteristics of their corresponding inputs. One class of such queries include identifying which $q$-quantile does each data point belong to. This constitutes a popular class of queries for domains such as annual employee salaries, annual sales figures of commercial firms and student test scores. For example, suppose the dataset consists of the annual sales figures of different clothing firms and the goal is to group them according to their respective deciles. Here, another class of queries, the partition can be defined directly on the input domain based on its semantics. Consider an example where the goal is to group a dataset of audiences of TV shows based on their age demographic – the domain of age can be divided into classes like "youth", "senior citizens". Once the partition is defined, each data owner uses ОРп to report their noisy encoding. Note that the dLDP guarantee is amenable to these cases, as one would want to report the intervals correctly but the adversary should not be able to distinguish between values belonging to the same interval.

### Frequency Estimation
Here, we discuss the default case of the ОРп primitive where the partition is same as the input domain, i.e., $\mathcal{P} = X$. Under this assumption, we can obtain a frequency oracle in the LDP setting under the dLDP guarantee. We describe the mechanism below (see Alg. 3 in the full paper [46] for full algorithm). Given a privacy parameter, $\epsilon$, each data owner, $DO_i, i \in [n]$, reports $\hat{o}_i = \text{OPec}(x_i, X, \epsilon)$ to the untrusted data aggregator. Next, the data aggregator performs non-negative least squares (NNLS) as a post-processing inferencing step on the noisy data to compute the final frequency estimations. NNLS is a type of constrained least squares optimizations problem where the coefficients are not allowed to become negative. That is, given a matrix $A$ and a (column) vector of response variables $Y$, the goal is to find $X$ such that

$$
\text{arg min}_{X} \|A \cdot X - Y\|_2, \text{subject to } X \geq 0
$$

where $\| \cdot \|_2$ denotes Euclidean norm. The rationale behind this inferencing step is discussed below.

**Lemma 3.** W.l.o.g let $X = \{1, \ldots , m\}$ and let $Y$ be the vector such that $Y(i), i \in [m]$ indicates the count of value $i$ in the set $\{\hat{o}_1, \ldots , \hat{o}_n\}$ where $\hat{o}_i = \text{OPec}(x_i, X, \epsilon)$. Given,

$$
A(i, j) = \text{Pr}[\text{OPec}(i, X, \epsilon) = j], i, j \in [m]
$$

the solution $X$ of $A \cdot X = Y$ gives an unbiased frequency estimator ($X(i)$ is the unbiased estimator for value $i$).

The proof of the above lemma is presented in the full paper [46]. Thus by the above lemma, $X$ is an unbiased frequency estimator. However, it is important to note that the solution $X$ is not guaranteed to be non-negative. But, given our problem setting, the count estimates are constrained to be non-negative. Hence, we opt for an NNLS inferencing. When the exact solution $X = A^{-1} \cdot Y$ is itself non-negative, the estimator obtained from the NNLS optimization is identical to the exact solution. Otherwise, the NNLS optimization gives a biased non-negative estimator that results in minimal least square error. The resulting frequency oracle can be used to answer other queries like mean estimation and range queries$^4$. A formal utility analysis is in the full paper [46].

### 4 $\epsilon$-dDP ORDER PRESERVING ENCRYPTION (ОРп)
In this section, we describe our proposed $\epsilon$-dDP order preserving encryption scheme, ОРп.

#### 4.1 Definition of ОРп
The $\epsilon$-dDP order preserving encryption (ОРп) scheme is an encryption scheme that bolsters the cryptographic guarantee of a ОРп scheme with an additional dDP guarantee. Here, we detail how our proposed primitive ОРп can be used in conjunction with a ОПе scheme (Def. 2.5) to form a ОРп scheme.

**Definition 4.1** ($\epsilon$-dDP Order Preserving Encryption, ОРп). A $\epsilon$-dDP order preserving encryption scheme, ОРп, is composed of a ОПе scheme, ОПе, that satisfies the IND-FA-OCPA guarantee (Def. 4.2), and the ОРп primitive and is defined by the following algorithms:

**ОРп Scheme**
- **Key Generation** ($K\epsilon$). Uses $K$ from the ОПе scheme to generate a secret key $S$.
- **Encryption** ($E\epsilon$). The encryption algorithm inputs a plaintext $x \in X$, an order $\Gamma$, a partition $\mathcal{P} \in \hat{X}$, and the privacy parameter $\epsilon$. It outputs $(S', y) \leftarrow E(S, \hat{o}, \Gamma)$ where $\hat{o} \leftarrow \text{OPec}(x, \mathcal{P}, \epsilon/2)$.
- **Decryption** ($D\epsilon$). The decryption algorithm uses $D$ to get back $\hat{o} \leftarrow D(S, y)$.

Following the above definition, the encryption of a dataset $X \in X^\omega, X = \langle x_1, \ldots , x_n\rangle$ is carried out as follows:

1. Set $S_0 \leftarrow K(1^\epsilon)$
2. For $i \in [n]$, compute $(S_i, y_i) \leftarrow E(S_{i-1}, \hat{o}_i, \Gamma)$ where $\hat{o}_i \leftarrow \text{OPec}(x_i, \mathcal{P}, \epsilon/2)$

**Key Idea.** A ОРп scheme works as follows:
- First, obtain an (randomized) encoding for the input using the ОРп primitive (one possible construction is given by Alg. 1 for any given $\epsilon$ and partition $\mathcal{P}$).
- Encrypt the above encoding with a ОПе scheme.

---

$^4$In the LDP setting, this refers to statistical range query, i.e., the count of the records that belong to a queried range.
Thus, ciphertexts encrypted with OPE preserve the order of the corresponding encodings as output by the OPE primitive. Referring back to our example, if $X = \{76, 9, 40, 15, 76, 77\}$ and its corresponding encodings are $\hat{O} = \{8, 1, 2, 4, 2, 8, 8\}$, then the ciphertext of $X$ under OPE preserves the order of $\hat{O}$.

In other words, since a OPE scheme preserves the exact order of its input dataset by definition, the utility of OPE (in terms of the preserved ordering information) is determined by the underlying OPE primitive. This is formalized by the following theorem.

**Theorem 4 (Utility Theorem).** If, for a given partition $\mathcal{P} \in \hat{X}$ and for all $x, x' \in \hat{X}$ such that $x > x'$ we have

$$\Pr[\text{OPE}(x, \mathcal{P}, e) \geq \text{OPE}(x', \mathcal{P}, e)] \geq \alpha, \alpha \in [0, 1] \quad (9)$$

then for a OPE scheme instantiated on such a OPE primitive,

$$\Pr[\mathcal{E}(x, S, \Gamma, \mathcal{P}, e) \geq \mathcal{E}(x', S, \Gamma, \mathcal{P}, e)] \geq \alpha \quad (10)$$

where $S \leftarrow \mathcal{K}_e(1^n)$ and any $\Gamma$.

The proof follows directly from Defs. 2.5 and 4.1.

**Lemma 5.** OPE satisfies $\frac{1}{2}$-dLDP.

The proof of the above lemma follows trivially from the post-processing guarantee of dLDP (Thm. 1).

### 4.2 New Security Definition for OPE

Here, we present a novel security guarantee for OPE, namely *indistinguishability under frequency-analyzing $\epsilon$-dDP ordered chosen plaintext attacks* (e-IND-FA-OCPA, Def. 4.2). We start with an intuitive explanation of the e-IND-FA-OCPA guarantee and its associated security game $G_{\text{IND-FA-OCPA}}$ followed by an example (Example 7) for intuitive explanation.

The e-IND-FA-OCPA guarantee is associated with a security game, $G_{\text{IND-FA-OCPA}}$, where the adversary, $\mathcal{A}_{\text{IND-FA-OCPA}}$, first chooses four input dataset of equal length, $X_{00}, X_{01}, X_{10}$ and $X_{11}$, such that $\mathcal{P}_0(X_{00})$ and $\mathcal{P}_1(X_{10})$ share at least one randomized order where $X_{00}, X_{01} \in X_0^n$, $X_{10}, X_{11} \in X_1^n$, $\mathcal{P}_0 \in \hat{X}_0$ and $\mathcal{P}_1 \in \hat{X}_1$. Additionally, $(X_{00}, X_{01})$ and $(X_{10}, X_{11})$ are $t$-adjacent (Def. 2.4). The challenger then selects two bits $\{b_1, b_2\}$ uniformly at random and returns the corresponding ciphertext for the dataset $X_{b_1, b_2}$. $\mathcal{A}_{\text{IND-FA-OCPA}}$ then outputs their guess for the bits and wins the game if they are able to guess either of the bits successfully. The e-IND-FA-OCPA guarantee states that $\mathcal{A}_{\text{IND-FA-OCPA}}$ cannot distinguish among the four datasets. In what follows, we first present its formal definition and then, illustrate it using an example.

**Definition 4.2 (e-IND-FA-OCPA).** An encryption scheme $E_{\text{e}} = (K_{\text{e}}, E_{\text{e}}, D_{\text{e}})$ has indistinguishable ciphertexts under frequency-analyzing $\epsilon$-dDP ordered chosen plaintext attacks if for any PPT adversary, $\mathcal{A}_{\text{IND-FA-OCPA}}$, and security parameter, $\kappa$:

$$\Pr[\text{Game } G_{\text{IND-FA-OCPA}}(\kappa, b_1, b_2)] = (c_1, c_2) \leq e^{\epsilon \kappa} \cdot \Pr[\text{Game } G_{\text{IND-FA-OCPA}}(\kappa, b_1', b_2') = (c_1, c_2)] + \text{negl}(\kappa) \quad (11)$$

where $b_1, b_2, b_1', b_2', c_1, c_2 \in \{0, 1\}$ and $G_{\text{IND-FA-OCPA}}(\kappa, b_1, b_2)$ is the random variable indicating the adversary $\mathcal{A}_{\text{IND-FA-OCPA}}$’s output for following security game:

**Game** $G_{\text{IND-FA-OCPA}}(\kappa, b_1, b_2)$

\begin{enumerate}
  \item $(X_{00}, X_{01}, X_{10}, X_{11}) \leftarrow \mathcal{A}_{\text{IND-FA-OCPA}}$
  \item $(a) \ X_{00}, X_{01} \in X_0^n$ and $X_{10}, X_{11} \in X_1^n$
  \item $\mathcal{P}_0(X_{00})$ and $\mathcal{P}_1(X_{10})$ have at least one common randomized order where $\mathcal{P}_0 \in \hat{X}_0$ and $\mathcal{P}_1 \in \hat{X}_1$
  \item $(X_{00}, X_{01})$ and $(X_{10}, X_{11})$ are $t$-adjacent (Def. 2.4)
  \item $\mathcal{O}_0 \leftarrow \mathcal{A}_{\text{IND-FA-OCPA}}(X_{00}, \mathcal{P}_0, \frac{\epsilon}{2})$
  \item $\mathcal{O}_1 \leftarrow \mathcal{A}_{\text{IND-FA-OCPA}}(X_{10}, \mathcal{P}_1, \frac{\epsilon}{2})$
  \item $(\mathcal{O}_0, \mathcal{O}_1)$ do not have any common randomized order, then return $\bot$. Else
  \item Select two uniform bits $b_1$ and $b_2$ and a randomized order $\Gamma^*$ common to both $\mathcal{O}_0$ and $\mathcal{O}_1$.
  \item If $b_2 = 0$, compute $Y_{b_1, b_2} \leftarrow E_{\text{e}}(\hat{O}_{b_1, \hat{S}, \Gamma^*, \mathcal{P}_b, \epsilon})^2$. Else, compute $Y_{b_1, b_2} \leftarrow E_{\text{e}}(X_{b_1, \hat{S}, \Gamma^*, \mathcal{P}_b, \epsilon})^2$.
  \item $(c_1, c_2) \leftarrow \mathcal{A}_{\text{IND-FA-OCPA}}(Y_{b_1, b_2})$ where $c_1$ and $c_2$ is $\mathcal{A}_{\text{IND-FA-OCPA}}$’s guess for $b_1$ (2b).
\end{enumerate}

$\mathcal{A}_{\text{IND-FA-OCPA}}$ is said to win the above game if $b_1 = c_1$ or $b_2 = c_2$.

**Example 7.** We illustrate the above definition using the following example. Consider $X_{00} = \{22, 94, 23, 94, 36, 95\}$, $X_{10} = \{9, 40, 11, 76, 15, 76\}$, $X_{01} = \{24, 94, 23, 94, 36, 95\}$ and $X_{11} = \{9, 40, 8, 76, 15, 76\}$ where $(X_{00}, X_{01})$ share a randomized order, $(1, 4, 2, 5, 3, 6)$, and $(X_{00}, X_{10})$ and $(X_{10}, X_{11})$ are $3$-adjacent. For the ease of understanding, we consider the default case of $\mathcal{P}_0 = \mathcal{O}_0 = X_0$ and $\mathcal{P}_1 = \mathcal{O}_1 = X_1$. This means that $\mathcal{P}_0(X_{00}) = X_{00}$ and so on.

**OPE.** If only OPE were to be used to encode the above datasets, then only the pairs $(X_{00}, X_{01})$ and $(X_{10}, X_{11})$ would be indistinguishable to the adversary (albeit an information theoretic one) because of the $\epsilon$-dDP guarantee (Defn. 2.4). However, there would be no formal guarantee on the pairs $(X_{01}, X_{11}), (X_{01}, X_{10}), (X_{00}, X_{11}), (X_{00}, X_{10})$.

**OPE.** Using OPE makes all 6 pairs $(X_{00}, X_{01}), (X_{00}, X_{11}), (X_{01}, X_{10}), (X_{10}, X_{11})$ indistinguishable for $\mathcal{A}_{\text{OPE}}$. This is because OPE essentially preserves the order of a $\epsilon$-dDP scheme.

Hence, OPE enjoys *strictly stronger security* than both OPE and OPE.

**Theorem 6.** The proposed encryption scheme, OPE satisfies $\epsilon$-IND-FA-OCPA security guarantee.

**Proof Sketch.** The proof of the above theorem follows directly from the IND-FA-OCPA guarantee of the OPE scheme and the fact that OPE satisfies $\epsilon/2$-dDP guarantee (Lemma 10 in the full paper [46]). The full proof is in the full paper [46].
Let \( N_G(X) = \{X' | X' \in X^n \text{ and } \{X, X'\} \text{ are indistinguishable} \} \). Additionally, we assume \( \mathcal{P} = X \) for the ease of understanding. Thus, in a nutshell, the \( \varepsilon \)-dDP guarantee allows a pair of datasets \( \{X, X'\} \) to be indistinguishable\(^6\) only if they are \( t \)-adjacent (for relatively small values of \( t \)). Referring back to our example, we have \( X_{01} \in N_{\varepsilon\text{-dDP}}(X_{00}) \) and \( X_{11} \in N_{\varepsilon\text{-dDP}}(X_{10}) \).

On the other hand, under the IND-FA-OCPA guarantee, \( \{X, X'\} \) is indistinguishable\(^7\) to \( \mathcal{A}_{\text{ppt}} \) only if they share a common randomized order. For instance, \( X_{00} \in N_{\text{IND-FA-OCPA}(X_{00})} \).

In addition to the above cases, the \( \varepsilon \)-IND-FA-OCPA guarantee allows a pair of datasets \( \{X, X'\} \) to be indistinguishable\(^8\) for \( \mathcal{A}_{\text{ppt}} \) if \( \{X, X'\} \)

- do not share a randomized order
- are not adjacent,

but there exists another dataset \( X'' \) such that

- \( \{X', X''\} \) are adjacent, i.e. \( X' \in N_{\text{dDP}}(X'') \)
- \( \{X, X''\} \) share a randomized order, i.e., \( X'' \in N_{\text{IND-FA-OCPA}}(X) \).

From our aforementioned example, we have \( X_{11} \not\in N_{\text{IND-FA-OCPA}}(X_{00}) \) and \( X_{10} \not\in N_{\text{IND-FA-OCPA}}(X_{10}) \) since \( X_{10} \in N_{\text{dDP}}(X_{10}) \) and \( X_{11} \in N_{\text{IND-FA-OCPA}}(X_{10}) \).

Thus, formally:

\[
N_{\text{IND-FA-OCPA}}(X) = \bigcup_{X'' \in N_{\text{dDP}}(X'')} N_{\text{IND-FA-OCPA}}(X) \tag{12}
\]

Since, trivially \( X \in N_{\text{IND-FA-OCPA}}(X) \) and \( X \in N_{\text{dDP}}(X) \), we have \( N_{\text{IND-FA-OCPA}}(X) \supseteq N_{\text{IND-FA-OCPA}}(X) \) and \( N_{\text{IND-FA-OCPA}}(X) \supseteq N_{\text{dDP}}(X) \).

**Key Insight.** The key insight of the \( \varepsilon \)-IND-FA-OCPA security guarantee is that the OPE scheme preserves the order of the outputs of a \( \varepsilon \)-dDP mechanism. As a result, the adversary is now restricted to only an \( \varepsilon \)-dDP order leakage from the ciphertexts. Hence, even if the security guarantee of the OPE layer is completely broken, the outputs of OPE would still satisfy \( \varepsilon \)-dDP due to Thm. 1. Referring to Example 1, in the very least input pairs \( \{X_{00}, X_{01}\} \) and \( \{X_{10}, X_{11}\} \) will remain indistinguishable under all inference attacks. Thus, OPE is the first encryption scheme to satisfy a formal security guarantee against all possible inference attacks and still provide some ordering information about the inputs.

**Remark 2.** The \( \varepsilon \)-IND-FA-OCPA guarantee of the OPE scheme is strictly stronger than both dDP (dLDP) and IND-FA-OCPA (the strongest possible guarantee for any OPE). Further, it depends only on the dLDP guarantee of the underlying OPE primitive which is independent of the partition \( \mathcal{P} \) used (as discussed in Sec. 3.2). We discuss the role of \( \mathcal{P} \) in Sec. 6.

### 5 OPE AND INFERENCE ATTACKS

In this section, we discuss the implications of OPE’s security guarantee in the face of inference attacks. Specifically, we formalize the protection provided by OPE’s (relaxed) DP guarantee – this is the worst case guarantee provided by OPE.

Recall that the \( \varepsilon \)-IND-FA-OCPA guarantee of an OPE bolsters the cryptographic guarantee of a OPE (IND-FA-OCPA) with an additional layer of (relaxed) DP guarantee. For the rest of the discussion, we focus on the worst case scenario where the OPE scheme provides no protection at all and study what formal guarantee we can achieve from just the (relaxed) DP guarantee. As discussed in Sec. 2.1, our proposed distance-based relaxation of DP comes in two flavors – local (dLDP, Def. 2.2) and central (dDP, Def. 2.4). Intuitively, dLDP is a guarantee for each individual data point while dDP is a guarantee for a dataset. As a refresher, Fig. 2 showcases the relationships between them. The most salient point is that the dLDP is a stronger guarantee than dDP – \( \varepsilon \)-dLDP implies \( \varepsilon \)-dDP. Thus, owing to the dLDP guarantee of the underlying OPE primitive, OPE trivially satisfies both dLDP (Lemma 5) and dDP (Lemma 10 in the full paper [46]) guarantees.

For our discussion in Sec. 4.2, we use the dDP guarantee since the IND-FA-OCPA guarantee of OPEs is also defined on datasets. In what follows, we show how to interpret the protection provided by OPE’s dLDP guarantee since it is stronger and holds for every data point. We do so with the help of an indistinguishability game, as is traditional for cryptographic security definitions. Let the input be drawn from a discrete domain of size \( N \), i.e., \( |X| = N \). The record indistinguishability game, \( G_{\beta \rightarrow \text{R1}} \), is characterized by a precision parameter \( \beta \in [\frac{1}{N}, 1] \). In this game, the adversary has to distinguish among a single record (data point) \( x \) and set of values \( Q(x) \) that differ from \( x \) by at most \( \beta N \). For instance, for \( x = 3, N = 10 \) and \( \beta = 1/5 \), the adversary has to distinguish among the values \( 3 \) and \( Q(3) = \{1, 2, 4, 5\} \). In the game, let \( y \) denote the ciphertext for \( x \) after encryption with OPE. The game is formally defined as follows:

**Game** \( G_{\beta \rightarrow \text{R1}}(p) \)

1. \( x_0 \leftarrow \mathcal{A} \)
2. \( Q(x) = \{x_1, \ldots, x_q\} \) where \( x_i \in X, i \in [q] \) s.t. \( |x_0 - x_i| \leq \beta N \) and \( x_i \neq x_0 \)
3. Select \( p \in \{0, 1, \ldots, q\} \) uniformly at random
4. \( y_p \leftarrow \mathcal{A}(y_p) \)

\( \mathcal{A} \) is said to win the above game if \( p' = p \). Let \( \text{rand} \) be a random variable indicating the output of the baseline strategy where the adversary just performs random guessing.

**Theorem 7.** For a OPE scheme satisfying \( \frac{1}{N} \)-dLDP, we have

\[
\left| \Pr[p' = p] - \Pr[\text{rand} = p] \right| \leq \frac{e^\varepsilon}{q + e^\varepsilon} - \frac{1}{q + 1} \tag{13}
\]

where \( \varepsilon = \frac{\beta N}{q} \) and \( q = \sqrt{Q(x_0)} \) (Step 2) of game \( G_{\beta \rightarrow \text{R1}} \).
From the above theorem, observe that for low values of $\epsilon^*$ (i.e., low $\epsilon$ and $\beta$) the R.H.S of the Eq. (30) is low. This means that for reasonably low values of $\epsilon$ (high privacy), with very high probability an adversary cannot distinguish among input values that are close to each other (small $\beta$) any better than just random guessing. Now, recall that owing to the dLDP guarantee of the underlying OPe primitive, every data point encrypted with OPe is also protected by the dLDP guarantee (Lemma 5). This implies that, for any dataset $X$, the above indistinguishability result holds for every individual data point (record) simultaneously. Thus, the dLDP guarantee rigorously limits the accuracy of any inference attack for every record of a dataset. The proof follows directly from the dLDP guarantee (see full paper [46]).

As a concrete example, let us look at the binomial attack [65] on OPE schemes satisfying IND-FA-OCPA. The attack uses a biased coin model to locate the range of ciphertexts corresponding to a particular plaintext. Experimental results on a dataset of first names show that the attack can recover records corresponding to certain high frequency plaintexts (such as, first name “Michael”) with high accuracy. In this context, the implications of the above result is as follows. Consider a dataset with plaintext records corresponding to first names “Michael” and “Michele”. For OPe, the recovery rate for either would not be better than the random guessing baseline since both the values are close to each other in alphabetic order.

Note that the above result is information-theoretic and holds for any adversary – active or passive, both in the persistent (access to volume/access-pattern/search-pattern leakage) and snapshot attack models (access to a single snapshot of the encrypted data) [55].

**Remark 4.** In the very least, OPe rigorously limits the accuracy of any inference attack for every record of a dataset for all adversaries. (Thm. 7).

### 6 OPe FOR ENCRYPTED DATABASES

In this section, we describe how to use a OPe scheme in practice for encrypted databases. We discuss how we can leverage the partition parameter, $P$, of the underlying OPe primitive for improved utility.

**Problem Setting.** For encrypted databases, a data owner has access to the entire database in the clear and encrypts it before outsourcing it to an untrusted server. The queriers of the encrypted databases are authorized entities with access to the secret keys. In many practical settings the data owner themselves is the querier [55].

The most popular use case for databases encrypted with OPeS is retrieving the set of records belonging to a queried range. However, due to randomization, encryption with OPe leads to loss in utility. Specifically in the context of range queries, it might miss some of the correct data records and return some incorrect ones. For the former, constraining OPe to maintain only a partial order is found to be helpful. As discussed in Sec. 3.2, the more coarse grained the partition is (the lesser the number of intervals), the larger is the probability for OPe to output the correct encoding. Hence, if any given range $[a, b]$ is covered by a relatively small number of intervals in $P$, then with high probability the set of records corresponding to the encodings $\{d | \tilde{d} \in O \land P(a) \leq \tilde{d} \leq P(b)\}$ will contain most of the correct records. This results in better accuracy for the subsequent OPe scheme since it’s accuracy is determined by the underlying OPe primitive (Thm. 4).

The problem of returning incorrect records can be mitigated by piggybacking every ciphertext encrypted with OPe with another ciphertext obtained from encrypting the corresponding plaintext with a standard authenticated encryption scheme [2], $\tilde{E} := (K, E, D)$. We refer to this as the augmented OPe scheme:

**Augmented OPe, $E^\dagger$**

- **Key Generation ($K_{E^\dagger}$).** This algorithm generates a pair of keys $\langle S, K \rangle$ where $S \leftarrow K_{E}(k)$ and $K \leftarrow K(k)$
- **Encryption ($E_{E^\dagger}$).** The encryption algorithm generates $\langle S', y_0, y_1 \rangle$ where $\tilde{y} \leftarrow \text{OPEc}(x, P, \epsilon/2), (S', y_0) \leftarrow E(S, \tilde{y}, \Gamma), y_1 \leftarrow \tilde{E}(K, x)$
- **Decryption ($D_{E^\dagger}$).** The decryption algorithm uses $S$ and $K$ to decrypt both the ciphertexts, $(x, \tilde{y})$ as $\tilde{y} \leftarrow D_2(S, y_0)$ and $x \leftarrow D_1(K, y_1)$.

After receiving the returned records from the server, the querier can decrypt $\{y_1\}$ and discard the irrelevant ones. The cost of this optimization for the querier is the processing overhead for the extra records (see discussion later).

For most input distributions, an equi-depth partitioning works well (as demonstrated by our evaluation in Sec. 7.2). Nevertheless, the partition can be updated dynamically as well (see the full paper [46] for details).

**Remark 5.** The partitioning of the input domain $(P)$ has no bearing on the formal security guarantee. It is performed completely from an utilitarian perspective in the context of encrypted databases – it results in an accuracy-overhead trade-off (accuracy = number of correct records retrieved; overhead = number of extra records processed).

**Range Query Protocol.** The end-to-end range query protocol is described in Alg. 2. Before detailing it, we will briefly discuss the protocol for answering range queries for a OPe scheme, $E$, that satisfies the IND-FA-OCPA guarantee (see [78] for details). Recall that every ciphertext is unique for such a OPe scheme. Hence, a querier has to maintain some state information for every plaintext. Specifically, if $Y = \{y_1, \ldots, y_n\}$ denotes the corresponding ciphertexts for an input set $X = \{x_1, \ldots, x_n\}$, then the querier stores the maximum and minimum ciphertext in $Y$ that corresponds to the plaintext $x_i$, denoted by $\text{max}_E(x_i)$ and $\text{min}_E(x_i)$, respectively. For answering a given range query $[a, b]$, the querier asks for all the records in $Y$ that belong to $[\text{min}_E(a), \text{max}_E(b)]$. Recall that in OPe, the OPe scheme is applied to the output (encodings) of the OPe primitive. So now for answering $[a, b]$, the querier has to retrieve records corresponding to $[P(a), P(b)]$ instead where $P$ is the partition for the encoding. Hence, the querier first maintains the state information for the encodings (Steps 1-6, Alg. 2). Note that since the size of the encoding space is smaller.

\*\*The augmented OPe scheme still upholds the $e$-IND-FA-OCPA guarantee owing to the semantic security of the encryption scheme $E$.\*\*
than the input domain $X$, the amount of state information to be stored for a OPE is less than that for a OPE (see full paper [46]). Next, the querier asks for all the encrypted records in the set $Y' = \{(y_{i0}, y_{i1})|i \in [n] \text{ and } y_{i0} \in [\min_{E_{\mathcal{P}}} (\mathcal{P}(a)), \max_{E_{\mathcal{P}}} (\mathcal{P}(b))]\}$ from the server (Steps 7-10). On receiving them, the querier only retains those records that fall in the queried range (Steps 11-18).

**Algorithm 2 Range Query Protocol**

Notations: $Z$ - Input dataset with $n$ records $(r_i, x_i)$ where $x_i \in X$ denotes the sensitive attribute to be encrypted under OPE and $r_i$ denotes the rest of associated data (other attributes could be encrypted as well);

- $S$ - Secret key for OPE; $\mathcal{P}$ - Partition for OPE;

$K$ - Secret key for the authenticated encryption scheme $\pi$

**Input:** Range Query $(a, b), a, b \in X$

**Output:** Set of records $V = \{r_i| (r_i, x_i) \in Z, x_i \in [a, b]\}$

**Initialization: Querier**

1. $X = (x_1, \ldots, x_n)$
2. $Y = E_{\pi}(X, S, K, \mathcal{P}, \pi, S)$ \(\triangleright\) Contains encrypted attributes \(\{(y_{i0}, y_{i1})\}\)
3. for $o \in O$
4. \quad $\max_{E_{\mathcal{P}}}(o) = \max \{y_{i0}| (y_{i0}, y_{i1}) \in Y \text{ and } y_{i0} \text{ decrypts to } o\}$
5. \quad $\min_{E_{\mathcal{P}}}(o) = \min \{y_{i0}| (y_{i0}, y_{i1}) \in Y \text{ and } y_{i0} \text{ decrypts to } o\}$
6. end for

**Range Query Protocol**

**Querier**

7. $C = (\min_{E_{\mathcal{P}}}(\mathcal{P}(a)), \max_{E_{\mathcal{P}}}(\mathcal{P}(b)))$ \(\triangleright\) Transformed range query based on state information

8. **Server** $\xrightarrow{C}$ **Querier**

9. $Y' = \{(y_{i0}, y_{i1})|i \in [n] \text{ and } y_{i0} \in [\min_{E_{\mathcal{P}}}(\mathcal{P}(a)), \max_{E_{\mathcal{P}}}(\mathcal{P}(b))]\}$

**Server** $\xrightarrow{Y'}$ **Querier**

10. for $y_{i1} \in Y'$
11. \quad $x_i' \leftarrow D(K, y_{i1})$
12. \quad if $(x_i' \in [a, b])$ \(\triangleright\) Verifying whether record falls in $[a, b]$
13. \quad $V = V \cup r_i$
14. \quad end if
15. end for
16. **Return** $V$

There are two ways the utility can be further improved. The first is including records from some of the intervals preceding $\mathcal{P}(a)$ and following $\mathcal{P}(b)$. The querier can ask for the records in $\{\max(o_1, o_{a-1}), \min(o_{a+1}, o_0)\}, l \in \mathbb{Z}_{\geq 0}$ where $o_a := \mathcal{P}(a)$ and $o_b := \mathcal{P}(b)$. However, the cost is an increased number of extra records.

Another optimization is to answer a workload of range queries at a time. Under OPE, queries can be made only at the granularity of the partition. Thus, if a queried range $[a, b]$ is much smaller than $[\mathcal{P}(a), \mathcal{P}(b)]$, then the querier has to pay the overhead of processing extra records. This cost can be reduced in the case of a workload of range queries where multiple queries fall within $[\mathcal{P}(a), \mathcal{P}(b)]$ (records that are irrelevant for one query might be relevant for some other in the workload). Additionally, the number of missing records for the query $[a, b]$ is also reduced if the records from the neighboring intervals of $[\mathcal{P}(a), \mathcal{P}(b)]$ are also included in the response (owing to the other queries in the workload).

**Discussion.** As described above, the server side interface for range query protocols is the same for both OPE and a OPE scheme with the IND-FA-OCPA guarantee (with a nominal change to accommodate the extra ciphertexts $(y_{i1})$). The cost is the extra storage for $(y_{i1})$. However, in this age of cloud services, outsourced storage ceases to be a bottleneck [16].

The querier, on the other hand, needs to decrypt all the returned records (specifically, $[y_{i1}]$). However, decryption is in general an efficient operation. For instance, the encryption of 1 million ciphertexts encrypted with AES-256 GCM requires < 3 minutes in our experimental setup. Thus, on the overall there is no tangible overhead in adopting OPE.

**Remark 6.** OPE could be used for secure data analytics in settings where (1) the $\epsilon$-DP guarantee is acceptable, i.e., the main security concern is preventing the distinction between input values close to each other, and (2) the application can tolerate a small loss in utility. Specifically in such settings, replacing encrypted databases that are already deploying OPE schemes (satisfying IND-FA-OCPA) with an OPE scheme would give a strictly stronger security guarantee against all attacks with nominal change in infrastructure or performance – a win-win situation.

7 EXPERIMENTAL EVALUATION

In this section, we present our evaluation results. Specifically, we answer the following three questions:

- **Q1:** Does OPE retrieve the queried records with high accuracy?
- **Q2:** Is the processing overhead of OPE reasonable?
- **Q3:** Can OPE answer statistical queries in the LDP setting with high accuracy?

**Evaluation Highlights**

- OPE retrieves almost all the records of the queried range. For instance, OPE only misses around 4 in every 10K correct records on average for a dataset of size ~ 732K with an attribute of domain size ~ 18K and $\epsilon = 1$.
- The overhead of processing the extra records for OPE is low. For example, for the above dataset, the number of extra records processed is just 0.3% of the dataset size for $\epsilon = 1$.
- We give an illustration of OPE’s protection against inference attacks. For an age dataset and an adversary with real-world auxiliary knowledge, no inference attack in the snapshot attack model can distinguish between two age values $(x, x')$ such that $|x - x'| \leq 8$ for $\epsilon = 0.1$.
- OPE can answer several queries in the LDP setting with high accuracy. For instance, OPE can answer ordinal queries with 94.5% accuracy for a dataset of size ~ 38K, an attribute of domain size ~ 240K and $\epsilon = 1$. Additionally, OPE achieves $6x$ lower error than the state-of-the-art $\epsilon$-LDP technique for frequency estimation for $\epsilon = 0.1$.

10$V$ has a small utility loss as explained before.
7.1 Experimental Setup

Datasets. We use the following datasets:

- **PUDF [5]**. This is a hospital discharge data from Texas. We use the 2013 PUDF data and the attribute PAT_ZIP (~ 732K records of patient’s 5-digit zipcode from the domain [70601, 88415]).
- **Statewide Planning and Research Cooperative System (SPARCS) [4]**. This is a hospital inpatient discharge dataset from the state of New York. This dataset has ~ 2532K records and we use the length_of_stay (domain [1, 120]) attribute for our experiments.
- **Salary [6]**. This dataset contains the compensation for San Francisco city employees. We use the attribute BasePay (domain [1000, 230000]) from the years 2011 (~ 36K records) and 2014 (~ 38K records).
- **Adult [19]**. This dataset is derived from the 1994 Census. The dataset has ~ 33K records and we use the attribute Age (domain [1, 100]) for our experiments.
- **Population [5]**. This is a US Census dataset of annual estimates of the resident population by age and sex. We use the data for male Puerto Ricans for the resident population by age and sex. We use the data for male Puerto Ricans for the years 2011 (~ 36K records) and 2014 (~ 38K records).

Metrics. We evaluate $Q_1$ using the relative percentage of missing records, $\rho_M = \frac{\text{missing records}}{\text{correct records}} \%$. Note that $\rho_M$ essentially captures false negatives which is the only type of error encountered – the querier can remove all cases false positives as discussed in Sec. 6. We evaluate $Q_2$ via the percentage of extra records processed relative to the dataset size, $\rho_E = \frac{\text{extra records}}{\text{records in dataset}} \%$. A key advantage of outsourcing is that the querier doesn’t have to store/process the entire database. $\rho_E$ measures this – low $\rho_E$ implies that the (relative) count of extra records is low and it is still advantageous to outsource. Thus, low $\rho_E$ implies that the client’s processing overhead is low (relative to the alternative of processing the whole dataset). This is a good metric for assessing the overhead because:

- For clients, the decryption of extra records doesn’t result in a tangible overhead (1 million records take < 3 minutes, Sec. 6).
- **OPe** has no impact on the server since its interface (functionality) is the same as that for OPes.

For evaluating ordinal queries (Fig. 4a), we use $\sigma_L = \%$ of points with $|P(x) - \delta_x| = k$ where $P(x)$ and $\delta_x$ denote the correct and noisy encoding for $x$, respectively. For instance, $\sigma_0 = 90$ means that 90% of the input data points were mapped to the correct bins. For frequency and mean estimation (Figs. 4b and 4c), we measure the absolute error $|c - \bar{c}|$ where $c$ is the true answer and $\bar{c}$ is the noisy output. For Fig. 4d, we use the error metric $|c - \bar{c}|/k$ where $k$ is the size of the query. We report the mean and std of error values over 100 repetitions for every experiment.

Configuration. The reported privacy parameter $\epsilon$ refers to the $\epsilon$-IND-FA-OCPA guarantee, and implies $\frac{\epsilon}{2} \cdot \text{DP}$ and $\frac{\epsilon}{2} \cdot \text{DLD}$ for OPes (Fig. 2). We instantiate the OPc primitive using Alg. 1. Due to lack of space, we present the results for all only two datasets, 1 dense (Adult, SPARCS) and 1 sparse (PUDF, Salary), in Fig. 3. The default settings are $\epsilon = 1$, equi-depth partitioning (based on a non-private prior) of sizes $|P| = 122$ for PUDF, $|P| = 8$ for SPARCS, $|P| = 10$ for Adult, and $|P| = 70$ for Salary. The range queries are chosen uniformly at random. We use the data from Salary for 2011 as an auxiliary dataset for Fig. 4a.

7.2 Experimental Results

7.2.1 Utility and Overhead of OPes. In this section, we evaluate $Q_1$ and $Q_2$ by computing the efficacy of OPes in retrieving the queried records of a range query. Recall, that a OPs scheme preserves the exact order of the plaintexts. Thus, the loss in accuracy (ordering information) arises solely from OPes’s use of the OPc primitive. Hence first, we study the effect of the parameters of OPes.

We start with the privacy parameter, $\epsilon$ (Figs. 3a and 3b). We observe that OPes achieves high utility even at high levels of privacy. For example, OPe misses only about 2% of the correct records (i.e., $\rho_M = 2\%$) for PUDF (Fig. 3a) on average even for a low value of $\epsilon = 0.01$ (i.e., the ratio of the output distributions of two datasets that are 100-adjacent is bounded by $\epsilon$). The associated processing overhead is also reasonable – the size of the extra records retrieved, $\rho_E$, is around 1% of the total dataset size on average. Next, we observe that as the value of $\epsilon$ increases, both the number of missing
and extra records drop. For instance, for \( \epsilon = 1 \) we have \( \rho_M = 0.04\% \), i.e., only 4 in every 10K correct records are missed on average. Additionally, the number of extra records processed is just 0.3% of the total dataset size on average. We observe similar trends for SPARC (Fig. 3b) as well. However, the utility for SPARC is lower than that for PUDF. For instance, \( \rho_M = 10\% \) and \( \rho_L = 35\% \) for \( \epsilon = 0.01 \) for SPARC. This is so because the ratio of the domain size (120) and partition size (8) for SPARC is smaller than that for PUDF (domain size \( \sim 18K \), \|\mathcal{P}\| = 122). As a result, the individual intervals for SPARC are relatively small which results in lower utility.

Next, we study the effect of the size of the partition (number of intervals) on OP\(\epsilon\)’s utility. As expected, we observe that for Adult, decreasing the partition size from 20 to 5 decreases \( \rho_M \) from 5% to 0.2% (Fig. 3c). However, this increases the number of extra records processed – \( \rho_L \) increases from 0.8% to 7%. Similar trends are observed for Salary (Fig. 3d).

Next, we study the effect of including neighboring intervals (Sec. 6) in Figs. 3e and 3f. For instance, for Salary, including records from 2 extra neighboring intervals drops \( \rho_M \) from 1% to 0%. However, \( \rho_L \) increases from 0.4% to 1.7%. The increase in \( \rho_L \) is more significant for Adult because the domain size for Adult is small and the dataset is dense. On the other hand, Salary has a larger and sparse domain.

Another way for improving utility is to answer a workload of range queries at a time (Sec. 6). We present the empirical results for this in Figs. 3g and 3h. For SPARC, we observe that \( \rho_M \) and \( \rho_L \) drop from 0.9% to 0.1% and 4% to 0.2%, respectively as the size of the workload is increased from 1 to 20. Further, we note that this effect is more pronounced for SPARC than for PUDF. This is because, the domain of PUDF is larger and hence, the probability that the queried ranges in the workload are close to each other is reduced.

7.2.2 Utility of OP\(\epsilon\) in the LDP Setting. In this section, we evaluate Q3 by studying the utility of the OP\(\epsilon\) primitive in the LDP setting.

First, we consider ordinal queries. For Adult, we define an equi-length partition \( \mathcal{P} = \{[1, 10], \ldots, [91, 100]\} \) over the domain and our query of interest is – Which age group (as defined by \( \mathcal{P} \)) does each data point belong to? For Salary, we define an equi-depth partition of size 10 over the domain and our query of interest is – Which decile does each data point belong to? Our results are reported in Fig. 4a. The first observation is that OP\(\epsilon\) reports the correct encodings with good accuracy. For instance, for \( \epsilon = 1 \), \( \sigma_0 = 94.5\% \) and \( \sigma_1 = 5.5\% \) for the Salary dataset. Another interesting observation is that for low values of \( \epsilon \), the accuracy for Salary is significantly higher than that for Adult. Specifically, for \( \epsilon = 0.1 \), \( \sigma_0 = 90\% \) and \( \sigma_1 = 35\% \) for Salary and Adult, respectively. The reason for this is two fold. Firstly, we use an auxiliary dataset for Salary to compute the weighted medians for the central tendencies. On the other hand, we do not use any auxiliary dataset for Adult and use the median of each interval as the central tendency. Secondly, the domain size of Salary (230K) is relatively large compared to the number of intervals (10) which results in higher utility (as explained in Sec. 3.2).

Fig. 4b shows our results for using OP\(\epsilon\) for frequency estimation. Baseline 1 denotes the state-of-the-art \( \epsilon \)-LDP frequency oracle [113]. We observe that OP\(\epsilon\) achieves significantly lower error than Baseline 1. For instance, the error of OP\(\epsilon\) is 6x lower than that of Baseline 1 for \( \epsilon = 0.1 \). This gain in accuracy is due to OP\(\epsilon\)’s relaxed \( \epsilon \)-LDP guarantee. From Fig. 4c, we observe that the frequency oracle designed via OP\(\epsilon\) can be used for mean estimation with high utility. Here Baseline 2 refers to the state-of-the-art protocol [49] for mean estimation for \( \epsilon \)-LDP. We observe that for \( \epsilon = 0.1 \), OP\(\epsilon\) achieves ~ 40x lower error than Baseline 2.

Another interesting observation is that OP\(\epsilon\)’s frequency oracle gives better accuracy for a range query of size \( k \) than \( k \) individual point queries (Fig. 4d). For instance, a range query of size 20 gives 5x lower error than 20 point queries. The reason behind this is that the output distribution of OP\(\epsilon\) is the exponential distribution centered at the input \( x \). Hence, with high probability \( x \) either gets mapped to itself or some other point in its proximity. Thus, the probability for accounting for most copies of \( x \) is higher for the case of answering a range query \( x \in [a, b] \) than for answering a point estimation for \( x \).

7.3 An Illustration of OP\(\epsilon\)’s Protection

Here, we give an illustration of OP\(\epsilon\)’s protection against inference attacks on a real-world dataset. We use the snapshot\(^1\) attack model (the adversary obtains a one-time copy or snapshot of the encrypted data [55]). Our analysis is based on a formal model that captures a generic inference attack in the snapshot model – we create a bitwise leakage profile for the plaintexts from the revealed order and adversary’s auxiliary knowledge as described below.

Model Description. We assume the input domain to be discrete and finite, and w.l.o.g denote it as \( X = [0, 2^{m-1}] \). Additionally, let \( D \) represent the true input distribution and \( X = \{x_1, \ldots, x_n\} \) be a dataset of size \( n \) with each data point sampled i.i.d from \( D \). We model our adversary, \( \mathcal{A}_{LDP} \), to have access to (1) auxiliary knowledge about a distribution, \( \mathcal{D}’ \), over the input domain, \( X \) and (2) the ciphertexts, \( C \), corresponding to \( X \) which represent the snapshot of the encrypted data store. The adversary’s goal is to

\(^1\)We use the snapshot attack for the ease of exposition; there are some caveats to its practicability [64].
recover as many bits of the plaintexts as possible. Let $X(i), i \in [n]$ represent the plaintext in $X$ with rank $[18] i$ and let $X(i), j, j \in [m]$ represent the $j$-th bit for $X(i)$. Additionally, let $b(i, j)$ represent the adversary’s guess for $X(i, j)$. Let $L$ be a $n \times m$ matrix where $L(i, j) = \Pr[X(i, j) = b(i, j) | D, D']$ represent the probability that $A_{\text{PPT}}$ correctly identifies the $j$-th bit of the plaintext with rank $i$. Hence, $L$ allows analysis of bitwise information leakage from $C$ to $A_{\text{PPT}}$. The rationale behind using this model is that it captures a generic inference attack in the snapshot model allowing analysis at the granularity of bits.

The exact computation of the bitwise leakage matrix, $L$, when $X$ is encrypted with $OPE$ and $OPE$ is given by Thms. 15 and 16, respectively, in the full paper [46]. Using these theorems, we analytically compute $L$ for a real-world dataset. We use the age data from the Population dataset for the year 2019 as the true input distribution, $D$, and the data from the year 2011 is considered to be the adversary’s auxiliary distribution, $D'$. We consider the dataset size to be 200 and the number of bits considered is 7 (domain of age is $[1, 100]$). Additionally, the partition $P$ for $OPE$ primitive is set to be $P = O = X = [1, 100]$. The reported privacy parameter $\epsilon$ refers to the $\epsilon$-IND-FA-OCPA guarantee, and implies $\epsilon$-dLDP and $\epsilon$-dDLP for $OPE$ (Fig. 2). As shown in Fig. 5, we observe that the probability of successfully recovering the plaintext bits is significantly lower for $OPE$ as compared to that of a $OPE$. Moreover, the probability of recovering the lower-order bits (bits in the right half) is lower than that of higher-order bits – the probability of recovering bits 7-5 is $\approx 0.5$ which is the random guessing baseline. Recall that Thm. 7 implies that the adversary would not be able to distinguish between pairs of inputs that are close to each other. Hence, the above observation is expected since values that are close to each other are most likely to differ only in the lower-order bits. Additionally, as expected, the probability of the adversary’s success decreases with decreasing value of $\epsilon$. For instance, the average probability of success for the adversary for bit 4 reduces from 0.77 in the case of $OPE$s (Fig. 5a) to 0.62 and 0.51 for $\epsilon = 1$ (Fig. 5b) and $\epsilon = 0.1$ (Fig. 5c), respectively, for $OPE$. Concretely, no inference attack in the snapshot model that uses the given auxiliary knowledge can distinguish between two age values $(x, x')$ such that $|x - x'| \leq 8$ for $\epsilon = 0.1$.

8 RELATED WORK

Relaxation of DP. dLDP is equivalent to metric-based LDP [24] where the metric used is $l_1$-norm. Further, metric-LDP is a generic form of Blowfish [69] and $d_1$-privacy [41] adapted to LDP. Other works [26, 37, 42, 67, 112, 114] have also considered metric spaces. DP and Cryptography. A growing number of work has been exploring the association between DP and cryptography. One such line of work proposes to allow a DP leakage of information for gaining efficiency in cryptographic protocols [28, 39, 59, 94, 110]. A parallel line of work involves efficient use of cryptographic primitives for differentially private functionalities [21, 33, 34, 44, 70, 104]. Additionally, recent works have combined DP and cryptography for distributed learning [22, 45, 76] (see full paper [46]).

9 CONCLUSION

We have proposed a novel $\epsilon$-dDLP order preserving encryption scheme, $OPE$. $OPE$ enjoys a formal guarantee of $\epsilon$-dDLP, in the least, even under inference attacks. To the best of our knowledge, this is the first work to combine DP with a property-preserving encryption scheme. Our experimental results show that $OPE$ and $OPE$ achieve high utility on real-world datasets.

REFERENCES

[1] 1992. TPCH Benchmark. http://www.tpc.org/tpch/.

[2] 2001. NIST: Block Cipher Techniques. https://csrc.nist.gov/projects/block-cipher-techniques/bcm/modes-development/.

[3] 2010. National Population by Characteristics: 2010-2019. https://www.census.gov/data/tables/time-series/demo/poptv/2010-national-detail.html.

[4] 2012. Hospital Inpatient Discharges. https://health.data.ny.gov/Health/Hospital-Inpatient-Discharges-SPARCS-De-Identified/utud-w5st/.

[5] 2013. Hospital Discharge Data Public Use Data File. http://www.dshs.state.tx.us/THHC/Hospitals/Download.shtml.

[6] 2015. SF Salaries, Kaggle. https://www.kaggle.com/kaggle/sf-salaries/.

[7] 2016. Anthem. Anthem data breach. https://www.anthemfacts.com/.

[8] 2016. Ciphercloud. https://www.ciphercloud.com/.

[9] 2016. Microsoft, Always Encrypted (Database Engine). https://msdn.microsoft.com/en-us/library/mt163865.aspx.

[10] 2016. Perspecsys: A Blue Coat Company. http://perspecsys.com/.

[11] 2016. Yahoo Data Breach. https://money.cnn.com/2016/09/22/technology/yahoo-data-breach/.

[12] 2017. Wikipedia. Sony pictures entertainment hack. https://en.wikipedia.org/wiki/Sony_Pictures_EntertainmentHack/.

[13] 2017. Wikipedia. Target customer privacy. https://en.wikipedia.org/wiki/Target_Corporations/CustomerPrivacy/.

[14] 2018. Wikipedia. Facebook–Cambridge Analytica data scandal. https://en.wikipedia.org/wiki/Facebook%E2%80%93Cambridge_Analytica_Data_scandal/.

[15] 2019. Facebook data breach. https://www.forbes.com/sites/daveywinder/2019/09/05/facebook-security-snafe-exposes-419-million-user-phone-numbers/.

[16] 2020. AWS Pricing. https://aws.amazon.com/s3/pricing/.

[17] 2020. Disclosure Avoidance and the 2020 Census. https://www.census.gov/about/policies/privacy/statistical_safeguards/disclosure-avoidance-2020-census.html.

[18] 2020. Ranking. https://en.wikipedia.org/wiki/Ranking/.

[19] A. Asuncion and D. Newman. 2010. UCI Machine Learning Repository.

[20] Jayadev Acharya, Keith Bonawitz, Peter Kairouz, Daniel Ramage, and Ziteng Sun. 2019. Context-Aware Local Differential Privacy. arXiv:1911.00038 [cs.LG].

[21] A. Agrawal, M. Heiby, S. Kamara, and Tariq Moataz. 2018. Encrypted Databases for Differential Privacy. Proceedings on Privacy Enhancing Technologies 2019 (2018), 170 – 190.

[22] Naman Agarwal, Ananda Theertha Suresh, Felix Xinnan X Yu, Sanjiv Kumar, and Brendan McMahan. 2018. csGDP. Communication-efficient and differentially-private distributed SGD. In Advances in Neural Information Processing Systems, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett (Eds.), Vol. 31. Curran Associates, Inc. https://proceedings.neurips.cc/paper/2018/file/21ce69121e39821d07d04fab832870-Paper.pdf.

[23] R. Agrawal, J. Kiernan, R. Srikant, and Y. Xu. 2004. Order Preserving Encryption for Numeric Data. In Proceedings of the 2004 ACM SIGMOD International Conference on Management of Data (Paris, France) (SIGMOD ’04). Association for Computing Machinery, New York, NY, USA, 563–574. https://doi.org/10.1145/1007568.1007632.

[24] M. Alvim, K. Chatzikokolakis, C. Palamidessi, and A. Pazii. 2018. Invited Paper: Orderaly Computing Techniques. https://csrc.nist.gov/projects/block-cipher-techniques/bcm/modes-development/.

[25] 2019. Facebook data breach. https://www.forbes.com/sites/daveywinder/2019/09/05/facebook-security-snafe-exposes-419-million-user-phone-numbers/.

[26] 2020. AWS Pricing. https://aws.amazon.com/s3/pricing/.

[27] R. Agrawal, J. Kiernan, R. Srikant, and Y. Xu. 2004. Order Preserving Encryption for Numeric Data. In Proceedings of the 2004 ACM SIGMOD Interna-

tional Conference on Management of Data (Paris, France) (SIGMOD ’04). As-sociation for Computing Machinery, New York, NY, USA, 563–574. https://

doi.org/10.1145/1007568.1007632.

[28] M. Alvim, K. Chatzikolokakis, C. Palamidessi, and A. Pazii. 2018. Invited Paper: Orderaly Computing Techniques. https://csrc.nist.gov/projects/block-cipher-techniques/bcm/modes-development/.

[29] M. Alvim, K. Chatzikokolakis, C. Palamidessi, and A. Pazii. 2018. Invited Paper: Orderaly Computing Techniques. https://csrc.nist.gov/projects/block-cipher-techniques/bcm/modes-development/.

[30] M. Alvim, K. Chatzikokolakis, C. Palamidessi, and A. Pazii. 2018. Invited Paper: Orderaly Computing Techniques. https://csrc.nist.gov/projects/block-cipher-techniques/bcm/modes-development/.

[31] M. Alvim, K. Chatzikokolakis, C. Palamidessi, and A. Pazii. 2018. Invited Paper: Orderaly Computing Techniques. https://csrc.nist.gov/projects/block-cipher-techniques/bcm/modes-development/.
Strengthening Order Preserving Encryption with Differential Privacy

CCS ‘22, November 7–11, 2022, Los Angeles, CA, USA