Secure and Utility-Aware Data Collection with Condensed Local Differential Privacy

Mehmet Emre Gursoy, *Student Member, IEEE,* Acar Tamersoy, Stacey Truex, Wenqi Wei, and Ling Liu, *Fellow, IEEE*

Abstract—Local Differential Privacy (LDP) is popularly used in practice for privacy-preserving data collection. Although existing LDP protocols offer high utility for large user populations (100,000 or more users), they perform poorly in scenarios with small user populations (such as those in the cybersecurity domain) and lack perturbation mechanisms that are effective for both ordinal and non-ordinal item sequences while protecting sequence length and content simultaneously. In this paper, we address the small user population problem by introducing the concept of Condensed Local Differential Privacy (CLDP) as a specialization of LDP, and develop a suite of CLDP protocols that offer desirable statistical utility while preserving privacy. Our protocols support different types of client data, ranging from ordinal data types in finite metric spaces (numeric malware infection statistics), to non-ordinal items (OS versions, transaction categories), and to sequences of ordinal and non-ordinal items. Extensive experiments are conducted on multiple datasets, including datasets that are an order of magnitude smaller than those used in existing approaches, which show that proposed CLDP protocols yield high utility. Furthermore, case studies with Symantec datasets demonstrate that our protocols accurately support key cybersecurity-focused tasks of detecting ransomware outbreaks, identifying targeted and vulnerable OSs, and inspecting suspicious activities on infected machines.

Index Terms—Privacy, local differential privacy, cybersecurity, malware epidemiology

◆

1 INTRODUCTION

Organizations and companies are becoming increasingly interested in collecting user data and telemetry to make data-driven decisions. While collecting and analyzing user data is beneficial to improve services and products, users' privacy poses a major concern. Recently, the concept of *Local Differential Privacy (LDP)* has emerged as the accepted standard for privacy-preserving data collection [1], [2], [3]. In LDP, each user locally perturbs their sensitive data on their device before sharing the perturbed version with the data collector. The perturbation is performed systematically such that the data collector cannot infer with strong confidence the true value of any user given their perturbed value, yet it can still make accurate inferences pertaining to the general population. Due to its desirable properties, LDP has been adopted by major companies to perform certain tasks, including Google to analyze browser homepages and default search engines in Chrome [2], [4], Apple for determining emoji frequencies and spelling prediction in iOS [5], [6], and Microsoft to collect application telemetry in Windows 10 [7].

While LDP is popularly used for the aforementioned purposes, one domain that is yet to embrace it is cybersecurity. It can be argued that this nuanced domain has the potential to greatly benefit from an LDP-like protection mechanism. This is because many security products rely on information collected from their clients, with the required telemetry ranging from file occurrence information in file reputation systems [8], [9], [10] to heterogeneous security event information such as system calls and memory

dumps in the context of Endpoint Detection and Response systems (see [11] for a survey on core behavioral detection techniques used by such systems). Nevertheless, clients are often reluctant to share such data fearing that it may reveal the applications they are running, the files they store, or the overall cyber hygiene of their devices. Offering formal privacy guarantees would help convince clients that sharing their data will not cause privacy leakages.

On the other hand, existing LDP protocols suffer from problems that hinder their deployment in the cybersecurity domain. First, although they are accurate for large populations (e.g., hundreds of thousands of clients), their accuracy suffers when client populations are smaller. Population size is not necessarily a problem for the likes of Google Chrome and Apple iOS with millions of active users, but it does cause problems in a domain such as cybersecurity where small population sizes are common. For example, if a security analyst is analyzing the behavior of a particular malware that targets a certain system or vulnerability, only those clients who are infected by the malware will have meaningful observations to report, but the number of infections could be limited to less than a couple of thousand users globally due to the targeted nature of the malware. In such cases, we need a new scheme that allows the security analyst to make accurate inferences while simultaneously giving adequate privacy to end users. Second, existing protocols consider a limited set of primitive data types. To the best of our knowledge, currently no protocol supports perturbation of item sequences (with either ordinal or non-ordinal item domains) to offer privacy with respect to sequence length and content simultaneously. Sequences are more difficult to handle compared to the data types of singleton items or itemsets since they are not only high-dimensional, but also contain an ordering that must be preserved for tasks such as pattern mining. Yet, sequential data is ubiquitous in cybersecurity, e.g., security logs, network traffic data, file downloads, and so

---

- *M.E. Gursoy, S. Truex, W. Wei and L. Liu are with the College of Computing, Georgia Institute of Technology, Atlanta, GA. E-mail: {memregursoy, staceytruex, wenqiwei}@gatech.edu, ling.liu@cc.gatech.edu*
- *A. Tamersoy is with Symantec Research Labs, Culver City, CA. E-mail: acar_tamersoy@symantec.com*

forth are all examples of sequential data.

In this paper, we propose the notion of *Condensed LDP (CLDP)* and a suite of protocols satisfying CLDP to tackle these issues. In practice, CLDP is similar to LDP with the addition of a *condensation* aspect, i.e., during the process of perturbation, similar outputs are systematically favored compared to distant outputs using condensed probability. We design protocols satisfying CLDP for various types of data, including singletons and sequences of ordinal and non-ordinal items. We show that CLDP can be satisfied by a variant of the Exponential Mechanism [12], and employ this mechanism as a building block in our *Ordinal-CLDP*, *Item-CLDP*, and *Sequence-CLDP* protocols. Our methods are generic and can be easily applied for privacy-preserving data collection in multiple domains, including but not limited to cybersecurity.

We consider a Bayesian adversary model for privacy-preserving data collection, which enables us to establish a formal connection between LDP and CLDP and ensures that they provide the same protection against this common adversary. The Bayesian model measures the adversary's maximum posterior confidence (MPC) in predicting the user's true value over all possible inputs and outputs of a perturbation mechanism, where higher adversarial confidence implies lower privacy protection for users. Under this Bayesian model, we derive how the parameters of CLDP protocols can be selected to give as strong protection as LDP protocols. Experiments conducted using this parameter selection show that our CLDP protocols provide high utility by yielding accurate insights for population sizes that are an order of magnitude smaller than those currently assumed by state-of-the-art LDP approaches.

We also perform extensive experiments to evaluate the effectiveness of our protocols on real-world case studies and public datasets. Experiments show that proposed CLDP protocols enable accurate frequency estimation, heavy hitter identification, and pattern mining while achieving strong protection against the adversary model considered. Using data from Symantec, a major cybersecurity vendor, we show that CLDP can be used in practice for use cases involving ransomware outbreak detection, OS vulnerability analysis, and inspecting suspicious activities on infected machines. In contrast, existing LDP protocols can either not be applied to these problems or their application yields unacceptable accuracy loss. To the best of our knowledge, our work is the first to apply the concept of local differential privacy to the nuanced domain of cybersecurity.

2 BACKGROUND AND PROBLEM SETTING

We consider the data collection setting, where there are many clients (*users*) and an untrusted data collector (*server*). Each client possesses a secret value. The client's secret value can be an ordinal item (e.g., numeric value or integer), a categorical item, a non-ordinal item, or a sequence of items. The server wants to collect data from clients to derive useful insights; however, since the clients do not trust the server, they perturb their secrets locally on their device before sharing the perturbed version with the server. Randomized perturbation ensures that the server, having observed the perturbed data, cannot infer the true value of any one client with strong probability. At the same time, the scheme allows the server to derive useful insights from *aggregates* of perturbed data by recovering statistics pertaining to the general population.

In Section 2.1, we start by introducing the threat and Bayesian adversary models we consider for measuring privacy protection in the above data collection setting. In Section 2.2, we define LDP as the current solution to privacy-preserving data collection. In

Section 2.3, we describe the utility model and provide preliminary analysis that shows LDP's utility loss under small user populations. Section 2.4 contains our problem statement.

2.1 Adversary Model

We use a Bayesian approach to measure privacy in the local data collection setting. Our Bayesian adversary model is similar to the Bayesian adversary formulations for membership privacy and location privacy in [13], [14], [15], [16]; but differs in the sense that the main goal of local privacy schemes is to offer confidentiality and plausible deniability for each individual user's secret. Higher plausible deniability implies lower adversarial prediction confidence and consequently higher privacy protection.

The threat stems from an untrusted third party (e.g., the data collector) inferring the true value of the user with high confidence from the perturbed value (s)he observes. Then, the goal of randomized perturbation is to stop an adversary $\mathcal{A}$, even if they can fully observe perturbed output y, from inferring the user's true secret v. Given y, the optimal attack strategy for $\mathcal{A}$ is:

$$\mathcal{A}(y) = \arg\max_{\hat{v} \in \mathcal{U}} \Pr[\hat{v}|y] \quad (1)$$

$$= \arg\max_{\hat{v} \in \mathcal{U}} \frac{\pi(\hat{v}) \cdot \Pr[f(\hat{v}) = y]}{\sum_{z \in \mathcal{U}} \pi(z) \cdot \Pr[f(z) = y]} \quad (2)$$

where $\pi(\hat{v})$ denotes the prior probability of $\hat{v}$, f denotes a perturbation function, and $\mathcal{U}$ denotes the universe of possible items. Then, worst-case privacy disclosure can be measured using the maximum posterior confidence (MPC) the adversary can achieve over all possible inputs and outputs:

$$\text{MPC} = \max_{v \in \mathcal{U}, y} \Pr[v|y] = \max_{v \in \mathcal{U}, y} \frac{\pi(v) \cdot \Pr[f(v) = y]}{\sum_{z \in \mathcal{U}} \pi(z) \cdot \Pr[f(z) = y]} \quad (3)$$

This establishes a mathematical framework under which we can quantify worst-case adversarial disclosure of perturbation protocols, for both informed adversaries (with prior knowledge π) and uninformed adversaries (e.g., by canceling out the π term, or equivalently, setting $\pi(x) = 1/|\mathcal{U}|$ for all $x \in \mathcal{U}$). When MPC is high, the adversary can more easily predict the true input of the user, hence f yields lower privacy. Thus, lower MPC is desired.

2.2 Local Differential Privacy

The state-of-the-art scheme currently used and deployed by major companies such as Google, Apple, and Microsoft in the data collection setting is LDP [2], [4], [5], [6], [7]. In LDP, each user perturbs their true value v using an algorithm Ψ and sends $\Psi(v)$ to the server. LDP can be formalized as follows.

Definition 1 (ε-LDP). A randomized algorithm Ψ satisfies ε-local differential privacy (ε-LDP), where $\varepsilon > 0$, if and only if for any inputs v_1, v_2 in universe $\mathcal{U}$, we have:

$$\forall y \in Range(\Psi): \ \frac{\Pr[\Psi(v_1) = y]}{\Pr[\Psi(v_2) = y]} \le e^{\varepsilon}$$

where $Range(\Psi)$ denotes the set of all possible outputs of algorithm Ψ.

Here, ε is the privacy parameter controlling the level of indistinguishability. Lower ε yields higher privacy. Several works were devoted to building accurate protocols that satisfy ε-LDP. These works were analyzed and compared in [3], and it was found that the two currently optimal protocols are based on: (i) OLH, the hashing extension of the GRR primitive and (ii) RAPPOR, the

Bloom filter-based bitvector encoding and bit flipping strategy. Next, we briefly present these protocols.

Generalized Randomized Response (GRR). This protocol is a generalization of the *Randomized Response* survey technique introduced in [17]. Given the user's true value v, the perturbation function Ψ_{GRR} outputs y with probability:

$$\Pr[\Psi_{GRR}(v) = y] = \begin{cases} p = \frac{e^{\varepsilon}}{e^{\varepsilon} + |\mathcal{U}| - 1} & \text{if } y = v \\ q = \frac{1}{e^{\varepsilon} + |\mathcal{U}| - 1} & \text{if } y \neq v \end{cases}$$

where $|\mathcal{U}|$ denotes the size of the universe. This satisfies ε-LDP since $\frac{p}{q} = e^{\varepsilon}$. In words, Ψ_{GRR} takes as input v and assigns a higher probability p to returning the same output $y = v$. With remaining $1 - p$ probability, Ψ_{GRR} samples a fake item from the universe $\mathcal{U} \setminus \{v\}$ uniformly at random, and outputs this fake item.

Optimized Local Hashing (OLH). When the universe size $|\mathcal{U}|$ is large, it dominates the denominator of p and q, thus the accuracy of GRR deteriorates quickly. The OLH protocol proposed in [3] handles the large universe problem by first using a hash function to map v into a smaller domain of hash values and then applying GRR on the hashed value. Formally, the client reports:

$$\Psi_{OLH}(v) = \langle H, \Psi_{GRR}(H(v)) \rangle$$

where H is a randomly chosen hash function from a family of hash functions. Each hash function in the family maps $v \in \mathcal{U}$ to a domain $\{1...g\}$, where g denotes the size of the hashed domain, typically $g \ll |\mathcal{U}|$. We use $g = \lceil e^{\varepsilon} + 1 \rceil$ as the optimal value of g found in [3].

Randomized Aggregatable Privacy-Preserving Ordinal Response (RAPPOR). RAPPOR was developed by Google and is used in Chrome [2]. In RAPPOR, the user's true value v is encoded in a bitvector B. The straightforward method is to use one-hot encoding such that B is a length-$|\mathcal{U}|$ binary vector where the v'th position is 1 and the remaining positions are 0. When $|\mathcal{U}|$ is large, both communication cost and inaccuracy cause problems, hence RAPPOR uses Bloom filter encoding. Specifically, B is treated as a Bloom filter and a set of hash functions $\mathcal{H}$ is used to map v into a set of integer positions that must be set to 1. That is, $\forall H \in \mathcal{H}, B[H(v)] = 1$, and the remaining positions are 0.

After the encoding, RAPPOR uses perturbation function Ψ_{RAPPOR} on B to obtain perturbed bitvector B' as follows:

$$\Pr\left[B'[i] = \Psi_{RAPPOR}(B[i]) = 1\right] = \begin{cases} \frac{e^{\varepsilon/2\Delta}}{e^{\varepsilon/2\Delta} + 1} & \text{if } B[i] = 1 \\ \frac{1}{e^{\varepsilon/2\Delta} + 1} & \text{if } B[i] = 0 \end{cases}$$

where 2Δ is analogous to the notion of *sensitivity* in differential privacy [18], i.e., how many positions can change in neighboring bitvectors at most? In one-hot encoding, $\Delta = 1$; in Bloom filter encoding, $\Delta = |\mathcal{H}|$. Then, the perturbation process considers each position in B independently, and the existing bit $B[i]$ is either kept or flipped when creating $B'[i]$.

2.3 Utility Model and Analysis

The most common use of LDP is to enable the data collector learn *aggregate* population statistics from large collections of perturbed data. Much research has been invested in tasks such as frequency estimation (identify proportion of users who have a certain item) and heavy hitter discovery (identify popular items that are held by largest number of users) [3], [19], [20], [21]. Utility is measured by how closely the privately collected statistics resemble the actual statistics that would have been obtained if privacy was not applied.
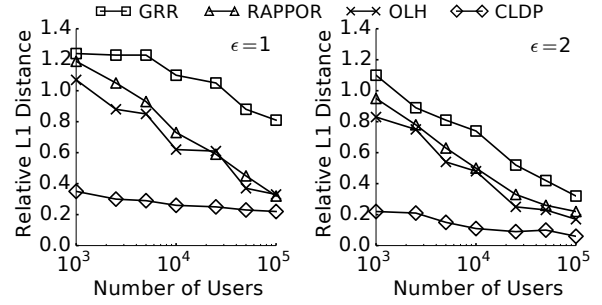

Fig. 1: Measuring relative error in item frequency estimation by calculating L1 distance between actual and estimated frequencies. L1 distance of d means $100 \cdot d$ % estimation error.

Frequency estimation and heavy hitter discovery are important tasks in the cybersecurity domain as well. For example, by monitoring the observed frequencies of different malware using aggregates of privatized malware reports, a cybersecurity vendor such as Symantec can identify large-scale malware outbreaks and create response teams to address them. Furthermore, analyzing the heavy hitter operating systems that are most commonly infected by the malware will enable Symantec understand OS vulnerabilities as well as the fraction of clients in its user base that are impacted. Relevant findings may also be used by Symantec when developing its next-generation anti-malware defenses.

A novel challenge posed by the cybersecurity domain, however, is accurately supporting small user populations. Typically, existing LDP literature assumes the availability of "large enough" user populations in the order of hundreds of thousands or millions of users [2], [3], [5], [21]. Yet, population sizes in the cybersecurity domain are typically much smaller. For example, consider a security analyst analyzing the behavior of a specific malware by studying infected user machines. It is often the case that malware targets a specific computing platform or software product, limiting the total number of infections to less than a couple of thousand users globally. We designed the frequency estimation experiment in Figure 1 to illustrate how the utility of existing LDP protocols suffer under such small populations. We sampled each user's secret value from a Gaussian distribution with mean $\mu = 50$ and standard deviation $\sigma = 12$, and rounded it to the nearest integer. The goal of the data collector is to estimate the true frequency of each integer. We run this experiment for varying number of users between 1,000 and 100,000 and graph the error in the frequency estimations made by the data collector.

We observe from Figure 1 that although more recent and optimized protocols improve previous ones by decreasing estimation error (e.g., OLH outperforms RAPPOR, which outperforms GRR); in cases with small user populations (e.g., 1000, 2500, or 5000 users) the improvements offered by more recent LDP protocols over previous ones are only 10-20%, whereas our proposed CLDP approach provides a remarkable 60-70% improvement. Furthermore, with 2500 users, estimation error is larger than 80% even for the state-of-the-art OLH algorithm, which is optimized for frequency estimation [3]. In contrast, our proposed CLDP solution is able to handle small user populations gracefully, with estimation errors lower than half of OLH's errors.

2.4 Problem Statement

As our analysis shows, low utility levels of existing LDP protocols under small user population sizes constitute an important obstacle towards their deployment in the cybersecurity domain.

This motivates us to seek alternative approaches and protocols for privacy-preserving data collection in this challenging scenario. The problem we study in this paper is to design a data collection scheme such that: (i) it gives at least as strong privacy protection as existing LDP protocols under the MPC adversary model, (ii) while doing so, it provides higher accuracy and data utility compared to existing protocols especially for small user populations, and (iii) it offers extensibility and generalizability to support complex data types such as different types of singleton items, itemsets, and sequences that are present in the cybersecurity domain.

3 PROPOSED SOLUTION

In this section, we introduce our proposed solution aproach to the problem stated above. We start with the notion of *Condensed* Local Differential Privacy (CLDP).

3.1 Condensed Local Differential Privacy

Let $\mathcal{U}$ denote the finite universe of possible values (items) and let $d : \mathcal{U} \times \mathcal{U} \to [0, \infty)$ be a distance function that takes as input two items $v_1, v_2 \in \mathcal{U}$ and measures their distance. We require d to satisfy the conditions for being a metric, i.e., non-negativity, symmetry, triangle inequality, and identity of discernibles. Then, CLDP can be formalized as follows.

Definition 2 (α-CLDP). A randomized algorithm Φ satisfies α-condensed local differential privacy (α-CLDP), where $\alpha > 0$, if and only if for any inputs $v_1, v_2 \in \mathcal{U}$:

$$\forall y \in Range(\Phi) : \quad \frac{\Pr[\Phi(v_1) = y]}{\Pr[\Phi(v_2) = y]} \le e^{\alpha \cdot d(v_1, v_2)}$$

where $Range(\Phi)$ denotes the set of all possible outputs of algorithm Φ.

LDP and CLDP follow a similar formalism, but differ in how their privacy parameters and indistinguishability properties work. Similar to ε-DP, α-CLDP satisfies the property that an adversary observing y will not be able to distinguish whether the original value was v_1 or v_2. However, in α-CLDP, indistinguishability is controlled also by items' distance $d(\cdot, \cdot)$ in addition to α. Consequently, as d increases, α must decrease to compensate, i.e., we have $\alpha \ll \varepsilon$. By definition, CLDP constitutes a metric-based extension of LDP. Metric-based extensions of differential privacy have been studied in the past under certain settings such as aggregate query answering in centralized statistical databases [22], geo-indistinguishability in location-based systems [23], [24], and protecting sensitive relationships between entities in graphs through k-edge differential privacy [25]. In contrast, we propose the metric-based CLDP extension in the *data collection* setting. Our data collection setting poses novel challenges due to: (i) the local privacy scenario, unlike centralized DP assumed in aggregate query answering and graph mining in which user data is collected in the clear first and privacy is applied after the data has been stored in a centralized database, (ii) data types that are different than tabular datasets, locations, and graphs, and (iii) establishing relationships and comparison between CLDP and LDP under the assumed adversary and utility models.

Since existing LDP protocols do not satisfy CLDP, we need new mechanisms and protocols supporting CLDP. We show below that a variant of the Exponential Mechanism (EM) [12] satisfies α-CLDP. EM is used in the remainder of the paper as a building block for more advanced CLDP protocols.

Exponential Mechanism (EM). Let $v \in \mathcal{U}$ be the user's true value, and let the Exponential Mechanism, denoted by Φ_{EM}, take as input v and output a perturbed value in $\mathcal{U}$, i.e., $\Phi_{EM} : \mathcal{U} \to \mathcal{U}$. Then, Φ_{EM} that produces output y with the following probability satisfies α-CLDP:

$$\forall y \in \mathcal{U} : \quad \Pr[\Phi_{EM}(v) = y] = \frac{e^{\frac{-\alpha \cdot d(v,y)}{2}}}{\sum_{z \in \mathcal{U}} e^{\frac{-\alpha \cdot d(v,z)}{2}}}$$

Theorem 1. Exponential Mechanism satisfies α-CLDP.

Proof. Provided in the supplementary material. $\square$

3.2 Privacy Protection of LDP and CLDP

We wish to find the appropriate α value to be used in CLDP so that the MPC under α-CLDP will be equal to or lower than the MPC under ε-LDP, which ensures that CLDP gives equal or better protection than LDP against the adversary we consider.

Question: Let $\mathcal{U}$, d and π be given. If there is an LDP protocol currently in place with privacy budget ε and we are interested in switching to α-CLDP, how should the value of α be selected to achieve equal or better protection than LDP according to the MPC adversary model?

Answer: Based on the quantification of the adversary's maximum posterior confidence in Equation 3, the requirement to have the MPC of CLDP less than or equal to that of LDP can be written as:

$$\max_{v,y} \frac{\pi(v) \cdot \Pr[\Phi(v) = y]}{\sum_{z \in \mathcal{U}} \pi(z) \cdot \Pr[\Phi(z) = y]} \le \max_{v,y} \frac{\pi(v) \cdot \Pr[\Psi(v) = y]}{\sum_{z \in \mathcal{U}} \pi(z) \cdot \Pr[\Psi(z) = y]} \tag{4}$$

where Ψ denotes LDP perturbation and Φ denotes CLDP perturbation. Using $\mathcal{U}$, d, π and ε, we can compute the right hand side, and then search for the largest α such that the left hand side remains smaller than the right hand side, iteratively by incrementing α in each iteration and re-computing the left hand side in each iteration. The complexity of computing the right hand side (left hand side is analogous) is $\mathcal{O}(|\mathcal{U}|^3) \cdot \mathcal{O}(\Psi)$, where $\mathcal{O}(\Psi)$ is the complexity of the perturbation mechanism, often linear or sublinear in $\mathcal{U}$ or constant time. This is assuming access and querying of $\pi(v)$ and $d(v_i, v_j)$ are constant time operations, as they can be pre-computed in $\mathcal{O}(|\mathcal{U}| + |\mathcal{U}|^2)$ time and stored in a form that supports efficient access, which does not affect overall complexity. In deployment, the computation to convert ε to α is a one-time cost performed at protocol setup time, therefore it does not have a negative impact on real-time user experience.

An alternative to solving Equation 4 could be to obtain a closed-form relationship between α and ε given $\mathcal{U}$, d, and π. We found that such a closed-form relationship can be established under particular cases such as when π is uniform or when d satisfies certain properties. However, the closed-form relationships require constructing and solving a high degree polynomial, which has higher execution time in practice than the iterative solution to Equation 4 which we presented above, due to the inherent hardness and time complexity of solving high degree polynomials. Hence, we give the current version of Equation 4 for wider practical applicability and more efficient implementation than the closed-form relationships we could derive. One of our ongoing future work directions is considering the derivation of efficient closed-form relationships.

Roles of Relevant Factors: Among the relevant factors, $\mathcal{U}$ impacts not only the outcome of ε to α conversion, but also the
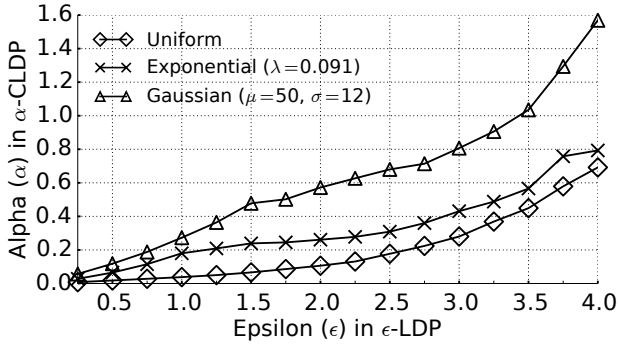
Fig. 2: Exemplifying the relationship between ε in ε-LDP and α in α-CLDP according to Equation 4, with $|\mathcal{U}| = 100$ and $\pi =$ Uniform, Exponential, Gaussian.

computational complexity. This is because $\mathcal{U}$ is a relevant factor in the $\Pr[\cdot]$ calculations in both LDP and CLDP. For example, the size of the bitvector in RAPPOR is affected by $|\mathcal{U}|$. EM of CLDP is also affected by $|\mathcal{U}|$, as each element in $\mathcal{U}$ must be assigned a score and the scores are then normalized. d has no impact on the behavior of LDP, but it affects CLDP. Since our conversion aims to achieve equivalent protection in CLDP compared to LDP against the Bayesian adversary from Section 2.1, it is expected that larger the $\max_{v_i, v_j} d(v_i, v_j)$, smaller the α value under the same ε. This can be explained via the fact that under fixed ε, according to the exponent in Definition 2, when d is larger, then α must be lowered to counter-balance the impact of increased d to still achieve equivalent protection. π is also a relevant factor in Equation 4. As we will exemplify in the upcoming practical analysis, when π is skewed (rather than uniform) it becomes the dominating factor in both the right and left hand sides of Equation 4; therefore under the same ε, typically larger α is obtained when π is skewed compared to uniform. Finally, the number of users does not appear in Equation 4 or play a role in the conversion, since the same ε is used across all users in LDP and similarly the same α is used across all users in CLDP.

Practical Analysis: To demonstrate the practicality of the relationship we establish above and derive insights, we solve Equation 4 under three example π settings. In all settings, we assume $\mathcal{U}$ is the set of integers between $[0, 99]$ and d measures absolute value distance between two integers. We use $\pi =$ Uniform, Gaussian and Exponential distributions with the corresponding distribution parameters given in Figure 2. Our rationale is that each π represents a different type of skewness: Uniform has no skewness, Gaussian is symmetrically skewed around the mean, and Exponential has positive skew. Users' secrets are samples from these distributions rounded to the nearest integer. Distribution parameters are chosen so that users' secrets fall within the universe of $[0, 99]$ with non-negligible tail probabilities. In Figure 2, we provide the results of our study for a wide range of ε values used in the literature: $0.25 \leq \varepsilon \leq 4$. Note that this figure is obtained purely by solving Equation 4, and does not require a real simulation or execution of the protocol involving end clients, i.e., the solution can be used in setup time before any data collection occurs.

Figure 2 allows us to derive several interesting insights. The first important conclusion is that ε and α are positively correlated – as the privacy requirement of LDP is relaxed, we can also relax that of CLDP. Second, it is often the case that LDP and CLDP give equivalent protection when $\alpha \ll \varepsilon$ holds. This is because in order to compensate for the added term of $d(\cdot, \cdot)$ decreasing

indistinguishability of distant items under CLDP, we have to use much smaller α in CLDP compared to ε of LDP. Third, we observe that the relationship between ε and α depends on π, e.g., if the data follows a Uniform distribution, CLDP must use a stricter α than the other two distributions. The reason is because there are two determining factors in calculating adversarial confidence: π and $\Pr[f(v) = y]$. For skewed distributions, π becomes the dominating factor and since the same π is shared by LDP and CLDP, the behavior of perturbation functions have relatively less impact on adversarial confidence. In contrast, for the Uniform distribution, since $\Pr[f(v) = y]$ becomes the dominating factor and its value is high for the tail-ends of the domain in the case of CLDP, we must use lower (stricter) α in CLDP to match the adversarial confidence in LDP. Based on our findings and observation that the Uniform π setting causes lowest α under the same ε compared to other π settings, in the remainder of the paper (including our experiments in Section 5) we assume the Uniform π setting. This helps demonstrate CLDP's utility benefit in the strictest and most challenging setting.

Note on Alternate Threat and Adversary Models: The analysis we perform in this section and the rest of the paper is under the Bayesian adversary and measurement of MPC described in Section 2.1. Since Bayesian methods were previously used in the literature for evaluating DP variants and relaxations in different domains, we argue that the Bayesian adversary model is relevant and representative [13], [14], [15], [16]. Yet, we acknowledge that under different adversaries, the protection offered by LDP and CLDP could be different, and thus, a different conversion between ε and α could be needed. For example, one alternative method to compare LDP and CLDP is hypothesis testing [26], [27], [28]. We leave comparison between LDP and CLDP under such different protection models to future work, since the main contribution of this paper is creating practical CLDP protocols and demonstrating their utility benefit under the Bayesian adversary model.

4 CLDP MECHANISMS AND PROTOCOLS

In this section, we present protocols that can be used in practice to collect data while achieving CLDP. We present three protocols: Ordinal-CLDP, Item-CLDP, and Sequence-CLDP, to address different types of client data.

4.1 Ordinal-CLDP for Ordinal Items

Our first protocol is Ordinal-CLDP, which addresses data types that stem from finite metric spaces, i.e., $\mathcal{U}$ is discrete and finite, and there exists a built-in distance metric $d : \mathcal{U} \times \mathcal{U} \to [0, \infty)$. This setting covers a variety of useful data types: (i) discrete numeric or integer domains where d can be the absolute value distance between two items, (ii) ordinal item domains with total order, e.g., letters and strings ordered by dictionary order A < B < C < ..., and (iii) categorical domains with tree-structured domain taxonomy where distance between two items can be measured using the depth of their most recent common ancestor in the taxonomy tree [29], [30]. In these scenarios, item order and d are naturally defined and enforced.

In Ordinal-CLDP, each client locally applies the Exponential Mechanism (EM) implementation shown in Algorithm 1, and uploads the perturbed output to the server. Notice that $\alpha, \mathcal{U}, d$, and user's true value v are all inputs to the algorithm. The algorithm's output v' is sent to the collector, thereby concluding the protocol in a single round without blocking.

Algorithm 1: Ordinal-CLDP using EM

Input : α: CLDP privacy parameter, $\mathcal{U}$: item universe,
　　　　d: distance metric, v: user's true value
Output: $v' \in \mathcal{U}$: perturbed value

1 for *each* $y \in \mathcal{U}$ **do**

2 $\quad$ Assign $\text{score}(y) = e^{\frac{-\alpha \cdot d(v,y)}{2}}$

3 end

4 Pick a random sample v' from $\mathcal{U}$, where
$\quad \Pr[v' \text{ is sampled}] = \frac{\text{score}(v')}{\sum_{z \in \mathcal{U}} \text{score}(z)}$

5 return v'

Algorithm 1 has some desirable utility properties. First, since d is a metric, it holds that $d(v,v) = 0$ for all $v \in \mathcal{U}$. As a result, each invocation of Algorithm 1 is unbiased since $\Pr[\Phi_{ORD}(v) = v] > \Pr[\Phi_{ORD}(v) = y]$ where $y \neq v$ and Φ_{ORD} denotes Algorithm 1. Furthermore, when d is selected with the property that $d(v_i, v_j) = c$ for all $v_i, v_j \in \mathcal{U}$ where $v_i \neq v_j$ and c is a constant such as $c = 1$, then the aggregation of the outputs of Algorithm 1 on the server side preserves the relative frequency relationship between v_i and v_j in expectation. In other words, if the true frequency of v_i is larger than v_j [resp. smaller than], after each client uploads perturbed items resulting from Algorithm 1 and the counts of the perturbed items are computed on the server side, the observed frequency of v_i is expected to be larger than the observed frequency of v_j [resp. smaller]. Given that frequency relationships between individual pairs of items are preserved, general item frequency rankings are also preserved. We provide the proof of this property in the supplementary material.

4.2 Item-CLDP for Non-Ordinal Items

Our second protocol is Item-CLDP in which each user still holds a singleton true item, but the items come from an arbitrary $\mathcal{U}$ with no pre-defined d or total order. For example, if $\mathcal{U}$ consists of OS names, an order of MacOS < Ubuntu < Windows is neither available nor initially justifiable. This non-ordinal item setting has been assumed in recent LDP research for finding popular emojis, emerging slang terms, popular and anomalous browser homepages, and merchant transactions [2], [3], [5], [6], [20]. We propose Item-CLDP in a generic way to maximize its scope and cover such existing cases. Parallel to previous works, our goal is to uphold relative item frequencies to learn popularity histograms and discover heavy hitters. To this end, we propose that a desirable perturbation strategy should replace a *popular* item with another popular item, and an *uncommon* item with another uncommon item. This achieves our goal of upholding relative item frequencies, as the expected behavior (conceptually) will be that popular items and uncommon items will be shuffled among themselves, and relative frequencies will be preserved.

The proposed Item-CLDP protocol is given in Figure 3. In Item-CLDP the server communicates with each client twice, hence the protocol consists of two rounds. The first round contains steps 1-3 and the second round contains steps 3-5. The server executes the first round with each client in parallel (without blocking). At the end of the first round, the server performs the aggregation and de-noising step (Step 3). Then, the server executes the second round of communication. Next, we explain each step in detail.

Step 1. When the protocol starts, the server knows universe $\mathcal{U}$ and each client has a true value v. The value of the privacy budget α and budget allocation parameter $0 < L < 1$ can be publicly

Client (v, α, L)　　　　　　Server $(\mathcal{U})$

Assume a random total order
in $\mathcal{U}$ such that $\{x_i\} \in \mathcal{U}$: $x_1 \leq$ ❶
$x_2 \leq \ldots \leq x_n$ and $d(x_{i,}\, x_{i+1}) = 1$

← Send $d, \mathcal{U}$

❷ Run $v' = $ Alg. 1 $(\alpha L, \mathcal{U}, d, v)$

Send v' →

Aggregate results from all clients,
apply denoising to infer refined d' ❸

← Send d'

❹ Run $v'' = $ Alg. 1 $(\alpha(1\text{-}L), \mathcal{U}, d', v)$

Send v'' →

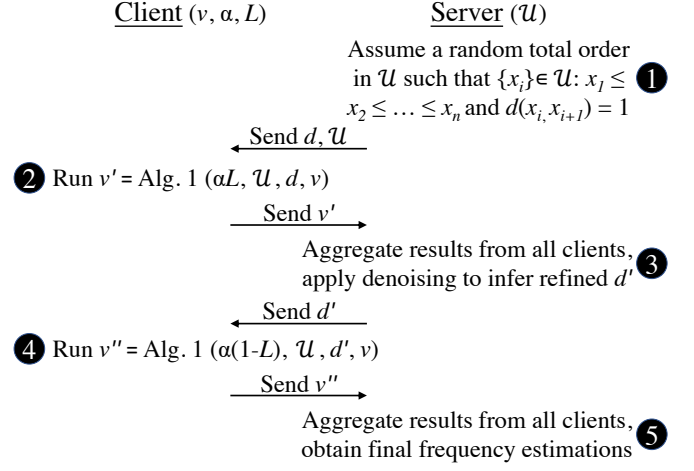Aggregate results from all clients, ❺
obtain final frequency estimations

Fig. 3: Item-CLDP protocol to report non-ordinal items.

known. (The role of L will be explained later.) A random total order is constructed among all items in $\mathcal{U}$ such that for $\{x_i\} \in \mathcal{U}$, $d(x_i, x_{i+1}) = 1$. The server advertises $\mathcal{U}$ and d to all clients.

Step 2. Each client runs Algorithm 1 with budget $\alpha \cdot L$ to obtain a perturbed value v' locally on their device. Then, the clients send their v' to the server.

Step 3. Due to the utility-unaware choice of d in Step 1, the absolute item frequencies discovered at this step contain significant error. A second round is desirable to reduce error. We found that although the server does not accurately learn absolute item frequencies by the end of Step 2, it can learn frequency *ranking* of items after applying a de-noising strategy. A key aspect is how de-noising is performed. Let $y \in \mathcal{U}$ be an item, let $\text{true}(y)$ be the true count of y in the population, which we are trying to find, and let $\text{obs}(y)$ denote the observed count of y following the first round of client-server communication (i.e., by the beginning of Step 3). The following holds in expectation:

$$\text{obs}(y) = \text{true}(y) \cdot \Pr[\Phi_{EM}(y) = y]$$
$$+ \sum_{x \in \mathcal{U} \setminus \{y\}} \text{true}(x) \cdot \Pr[\Phi_{EM}(x) = y]$$

Our goal is to solve for $\text{true}(y)$, but we cannot do so since $\text{true}(x)$ is also unknown. Hence, in our de-noising strategy we make the heuristic decision of plugging $\text{obs}(x)$ in place of $\text{true}(x)$, thereby obtaining $\text{true}'(y)$ as follows:

$$\text{true}'(y) = \frac{\text{obs}(y) - \sum_{x \in \mathcal{U} \setminus \{y\}} \text{obs}(x) \cdot \Pr[\Phi_{EM}(x) = y]}{\Pr[\Phi_{EM}(y) = y]}$$

We apply this to all y's in $\mathcal{U}$ and rank them according to their $\text{true}'(y)$. The distance function d' is set to reflect this new ranking instead of the original d from Step 1.

Step 4. After the clients receive d', each client runs Algorithm 1 with d' to obtain v'' and sends v'' to the server. This invocation of the algorithm is with budget $(1 - L) \cdot \alpha$.

Step 5. Upon receiving the v'' values from all clients, the server aggregates all results and obtains the final frequency estimates.

Since Item-CLDP is a two-round protocol, each client sends perturbed information twice. Hence, we need to quantify the total disclosure by the end of two rounds. We introduce the parameter L, which takes values between 0 and 1 and determines how the

CLDP privacy budget will be allocated to the two rounds of Item-CLDP. Denoting Item-CLDP by Φ_{ITEM}, we can show that for any two possible inputs v_1, v_2 of a user:

$$\frac{\Pr[\Phi_{ITEM}(v_1) = \langle v', v'' \rangle]}{\Pr[\Phi_{ITEM}(v_2) = \langle v', v'' \rangle]} \le e^{\alpha \cdot L \cdot d(v_1,v_2)} \cdot e^{\alpha \cdot (1-L) \cdot d'(v_1,v_2)}$$

$$\le e^{\alpha \cdot \max\{d(v_1,v_2), d'(v_1,v_2)\}}$$

This follows from the fact that the first round satisfies $(\alpha \cdot L)$-CLDP with d, and the second round satisfies $\alpha \cdot (1 - L)$-CLDP with d'. We choose the value of L by finding which L yields minimum frequency estimation error by the end of the second round of Item-CLDP. According to our experiments with different L, we recommend $L \cong 0.8$ as it often gives best results, which indicates that finding an accurate preliminary ranking in the first round of Item-CLDP is indeed important to obtain a good final result. An important property enabling the composition of Item-CLDP in above fashion is the fact that by construction of d and d' as metrics, $\max\{d(v_1, v_2), d'(v_1, v_2)\}$ is also a metric. This enables composition of the two individual CLDP rounds, and ensures their composed outcome achieves CLDP. The proof of composition is provided in the supplementary material.

4.3 Sequence-CLDP for Item Sequences

In Ordinal-CLDP and Item-CLDP, each user reports a single item. We now study the case where each user reports a collection of items. We give our Sequence-CLDP protocol assuming this collection forms a sequence and later show the applicability of Sequence-CLDP to set-valued data. Sequential data arises naturally in many domains, including cybersecurity (log files), genomics (DNA sequences), web browsing histories, and mobility traces; thus, a protocol for privacy-preserving collection of item sequences holds great practical value. We denote by X a user's true sequence, and by $X[i]$ the i'th element in X. Each element $X[i]$ is an item from universe $\mathcal{U}$. We assume the distance metric d between individual items is known apriori, e.g., for ordinal $\mathcal{U}$ we can use built-in d as in Ordinal-CLDP; otherwise, we can infer d using a process similar to the first round of Item-CLDP. We measure distance between two sequences $d_{\mathrm{seq}}(X, Y)$ as:

$$d_{\mathrm{seq}}(X,Y) = \sum_{i=1}^{|X|} d(X[i], Y[i])$$

In Sequence-CLDP, each client runs the sequence randomization procedure given in Algorithm 2 to locally perturb their X. The procedure has two probability parameters: $0 < halt, gen < 1$, and a length parameter *max_len* denoting the maximum sequence length allowed. Given true sequence X, the algorithm returns a perturbed sequence S. Our goal in Sequence-CLDP is to hide two complementary types of information: the *length* of X and the *contents* of X. For example, let X consist of a sequence of security events observed on a machine. Hiding the length of X is useful because it disables the adversary from learning that many security events were observed on this machine, hence that the machine is probably infected. Hiding the contents of X is useful because it disables the adversary from learning *which* security events were observed, hence the adversary cannot infer which types of problems exist on the machine, which attacks are successful, and so forth. Denoting Sequence-CLDP by Φ_{SEQ}, we formalize these privacy properties as follows.

Definition 3. Let $\Pr[\Phi_{SEQ}(X) \rightsquigarrow \ell]$ denote the probability that $\Phi_{SEQ}(X)$ produces a perturbed sequence of length ℓ given input sequence X. We say that Φ_{SEQ} satisfies α-length-indistinguishability if for any pair of true sequences X, Y:

$$\frac{\Pr[\Phi_{SEQ}(X) \rightsquigarrow \ell]}{\Pr[\Phi_{SEQ}(Y) \rightsquigarrow \ell]} \le e^{\alpha \cdot abs(|X|-|Y|)}$$

Definition 4. Let $\Pr[\Phi_{SEQ}(X) = S]$ denote the probability that $\Phi_{SEQ}(X)$ produces perturbed sequence S given input sequence X. We say that Φ_{SEQ} satisfies α-content-indistinguishability if, for any pair of true sequences X, Y of same length, it holds that:

$$\frac{\Pr[\Phi_{SEQ}(X) = S]}{\Pr[\Phi_{SEQ}(Y) = S]} \le e^{\alpha \cdot d_{\mathrm{seq}}(X,Y)}$$

It can be observed that both properties are adaptations of CLDP for hiding the two types of sensitive information relating to sequences: length and content. Definition 3 is analogous to CLDP with metric d being the absolute value difference between sequence lengths. It ensures that an adversary observing the length of the perturbed sequence ℓ cannot infer the length of the user's true sequence with high confidence. Definition 4 is analogous to CLDP with metric d being d_{seq}, i.e., item-wise difference in sequences' contents. It ensures that an adversary observing the contents of the perturbed sequence S cannot infer the contents of the user's true sequence with high confidence. The combined privacy protections offered by Definition 3 and Definition 4 can be best explained from the following perspective: Given that the adversary observes a perturbed sequence S with length $\ell = |S|$ that a user sends to the data collector, can the adversary reverse-engineer the length or the contents of the user's true sequence? If the perturbation satisfies Definitions 3 and 4 simultaneously, then the answer is negative, since both sequence length and content are protected analogously to the protection offered by CLDP.

Theorem 2. Algorithm 2 satisfies α-length-indistinguishability and α-content-indistinguishability simultaneously if *halt*, *gen* are selected either symmetrically as:

$$halt = gen = \frac{1}{e^{\alpha} + 1}$$

or asymmetrically within the ranges:

$$0 < halt < \frac{1}{e^{\alpha} + 1} \quad \text{and} \quad 1 - e^{\alpha} \cdot halt \le gen \le 1 - \frac{halt}{e^{\alpha}}$$

Proof. Provided in the supplementary material. $\square$

In Algorithm 2, high *halt* causes the algorithm to terminate early for a long sequence, causing the perturbed sequence S to be much shorter than X. High *gen* adds random items to S, thus it causes S to be much longer than X; in addition, since the added items are sampled uniformly at random, S will contain bogus elements. If we consider only the utility perspective, simultaneously decreasing the values of *halt* and *gen* yields higher sequence utility. However, Theorem 2 places bounds on the values of *halt* and *gen*; we cannot arbitrarily decrease them, otherwise we will not satisfy the indistinguishability properties. Among the given choices, asymmetric parameter choice is preferable when we expect users' true sequences to be long, since it assigns a lower halting probability compared to the symmetric case, thereby decreasing the probability that the algorithm is terminated early. This is done at the cost of increased *gen*, which implies that the asymmetric case will more likely add synthetic elements to S. Since this is detrimental to utility especially when users' true sequences are short, for short sequences, we recommend using the symmetric parameter choice.

Algorithm 2: Sequence-CLDP randomization

Parameter:	Probabilities *halt*, *gen*, maximum length allowed *max_len*
Input	: Client's private sequence X, privacy budget α, item universe $\mathcal{U}$, distance metric d
Output	: Randomized sequence S

1 **while** X *is shorter than max_len* **do**
2   Pad X with stop sign $\perp$ `// Dummy symbol`
3 **end**
4 Initialize empty sequence S
5 **for** $i = 1$ *to max_len* **do**
6   **if** $X[i] \neq \perp$ **then**
7   With probability *halt*, stop here and return S
8   With probability $(1 - halt)$, run Alg. 1 with inputs $(\alpha, \mathcal{U}, d, X[i])$ to obtain v', and append v' to S
9   **end**
10   **else**
11   With probability *gen*, randomly pick an item from $\mathcal{U}$ and append it to S
12   With probability $(1 - gen)$, stop here and return S
13   **end**
14 **end**
15 **return** S

Application to set-valued data. Although Sequence-CLDP is designed for sequences, it can be applied to set-valued data without information loss as follows. First, each user enforces a random ordering among the items in their itemset, to convert the itemset to a sequence. Second, the user runs Sequence-CLDP on this converted sequence to obtain a perturbed sequence. Third, the user removes the ordering from the perturbed sequence to obtain a perturbed itemset. Finally, the perturbed itemset is sent to the server. On the other hand, we cannot use existing set-valued LDP protocols [21], [31] on sequences without losing their sequentiality (ordering) aspect. Hence, we believe Sequence-CLDP has wider applicability than existing set-valued protocols.

5 EXPERIMENTAL EVALUATION

We compare our proposed CLDP protocols against the existing LDP protocols on real-world cybersecurity datasets provided by Symantec as well as on public datasets. In singleton item comparison, we use RAPPOR (proposed and deployed by Google [2]) and OLH (recent protocol with improved utility over prior works [3]). In set-valued setting, we use SVIM as the current state-of-the-art LDP protocol [31]. In each experimental dataset and setting, given $\mathcal{U}$, d and ε, we freshly execute Equation 4 and the process in Section 3.2 to obtain the appropriate α parameter for CLDP to ensure a fair comparison under each individual setting. Uniform π is used in each execution of Equation 4 since it is the most challenging π setting for CLDP to demonstrate its utility.

To ensure our experimental comparison between LDP and CLDP is fair with respect to the level of privacy protection under the Bayesian adversary model, we perform the following steps. First, we simulate data collection with LDP protocols and chosen ε, and measure the utility loss and privacy protection in terms of MPC. Second, we use the technique in Section 3.2 to determine the α that should be used for CLDP to match the protection achieved by LDP. Third, we simulate data collection with CLDP protocols and α from the previous step. Finally, we compare the utility loss caused by CLDP versus the utility loss of LDP.

In Section 5.1, we consider cybersecurity use cases that reflect the limitations of existing LDP protocols, e.g., user populations are small and sequential datasets cannot be handled. In these experiments, our results show that LDP protocols do not yield sufficient utility while our CLDP protocols offer satisfactory utility in most cases, hence their use in corresponding security products is practical and preferable. In Section 5.2, we experiment on public datasets which differ from the above since they do not reflect the limitations of LDP protocols, e.g., user populations are sufficiently large (over half a million). Even so, we show that our CLDP protocols perform at least comparable to, or in many cases better than, existing LDP protocols. An interesting result is that while LDP protocols are capable of finding the frequencies and ranking of heaviest hitter items with good accuracy (e.g., top 5-10% of the universe), CLDP protocols' accuracy is similar for these few heavy hitters, but they significantly outperform LDP protocols for remaining items (medium frequency and infrequent items).

5.1 Case Studies with Cybersecurity Datasets

Cybersecurity datasets provided by Symantec allowed us to test the accuracy of our protocols on pertinent real-world use cases and assess their practical applicability. We note that certain details such as the total number of infected machines are omitted on purpose due to confidentiality reasons.

Case Study #1: Ransomware Outbreak Detection

Setup. We consider the case where Symantec collects malware reports from machines running its anti-malware protection software. Each machine sends a locally private malware report to Symantec daily, containing the count of malware-related events observed on that machine during that day. Privacy is injected into malware reports by modifying the actual counts with LDP/CLDP.

For our experiments, we obtained the daily infection counts of two ransomware variants within those time periods in which we already know there were global outbreaks. Specifically, we considered the infections reported for *Cerber* between March 22 and April 21 in 2017 with the outbreak happening on April 6, and those reported for *Locky* between February 11 and March 13 in 2018 with the outbreak happening on February 26. We evaluated how accurately the total number of daily infections for these two ransomware variants can be estimated using our Ordinal-CLDP approach versus LDP approaches RAPPOR and OLH. The goal of our experiment is to retroactively test whether LDP/CLDP could identify if and when a ransomware outbreak happened.

Results. We illustrate the results in Figures 4 and 5 for Cerber and Locky respectively. If we use RAPPOR or OLH to perform detection, we obtain many false positives (days on which RAPPOR/OLH claim there was an outbreak, but in fact there was not) and false negatives (days on which RAPPOR/OLH claim there was no outbreak, but in fact there was). Some important examples are marked on the graphs, e.g., in Figure 4, RAPPOR raises false positives on March 23-24 as well as April 12-13. In addition, OLH misses the onset of the outbreak happening on April 5 by reporting 0 observed infections whereas in reality there are 16,388 infections. False positives are costly to Symantec since they cause the company to devote resources and response teams to combat a malware outbreak that does not exist. False negatives are also costly since Symantec will not react to an outbreak in a timely manner, losing customer trust. Observing this many false positives and false negatives with LDP methods raises serious concerns. In contrast, using Ordinal-CLDP, Symantec can obtain daily infection counts with high accuracy. Note that in Figures 4 and 5,
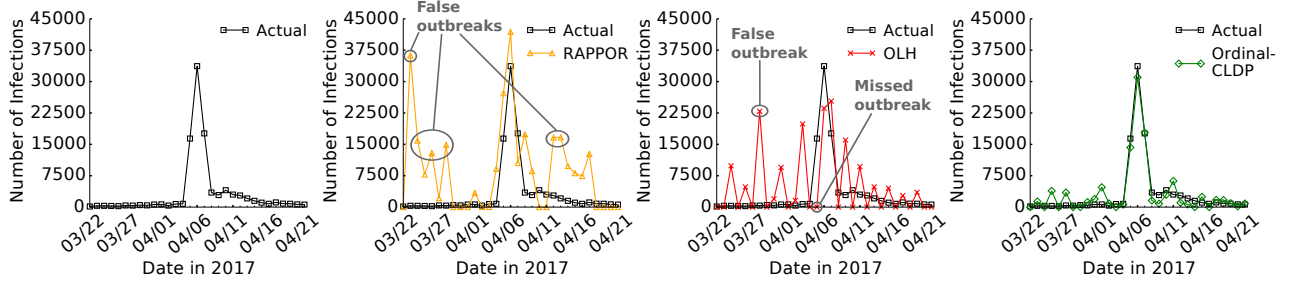
Fig. 4: Monitoring the infections of ransomware Cerber for one month to detect a potential outbreak. Illustrated in the graphs is the actual number of infections versus infections reported by LDP protocols RAPPOR and OLH, and our Ordinal-CLDP protocol ($\varepsilon = 1$). Ordinal-CLDP enables accurate recovery of daily infection counts and detection of outbreaks without major false positives or negatives.
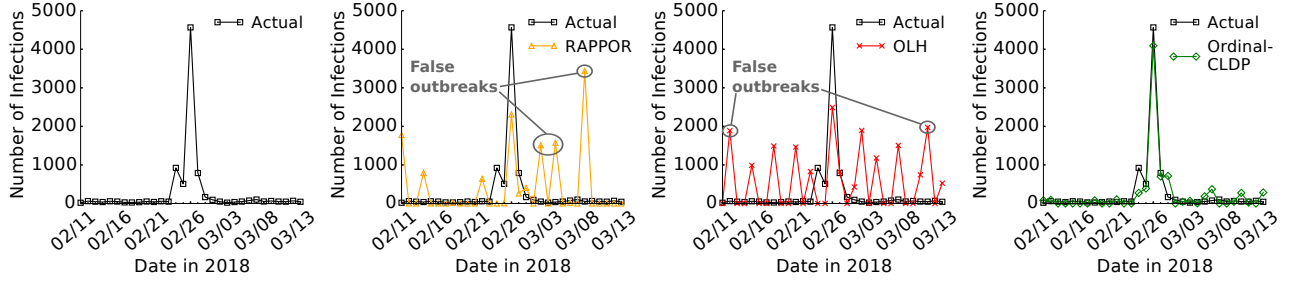

Fig. 5: Monitoring the infections of ransomware Locky for one month to detect a potential outbreak. Refer to Figure 4 for a similar experiment with a different ransomware ($\varepsilon = 1.0$). Same conclusions apply to this figure.

there are small discrepancies between the actual infections versus CLDP's predicted infections, which demonstrates that CLDP is not error-free. Nevertheless, using CLDP ransomware outbreaks can be detected in a privacy-preserving manner without major false positives or negatives.

Case Study #2: Ransomware Vulnerability Analysis

Setup. Next, we ask the question: Can we find which operating systems were most infected by ransomware? This would assist Symantec in discovering vulnerable or targeted OSs. When performing this analysis, we focus specifically on the day of outbreak (April 6, 2017 for Cerber and February 26, 2018 for Locky) and the machines reporting infections on this day. We assume Symantec obtains a locally private malware report from these machines including the vendor, specs, and OS version. Upon collecting reports from all machines, Symantec infers how frequently each OS was infected in the population, and ranks OSs in terms of infection frequency.

We conduct two experiments. In the first experiment, we find the actual (non-private) infection frequency of each OS, and compare actual frequencies with the frequencies that would be obtained if RAPPOR, OLH, or Item-CLDP were applied, using L1 distance as measurement of error. We vary ε between $0.5 \leq \varepsilon \leq 4$. In the second experiment, we fix $\varepsilon = 1$, rank the OSs in terms of infection frequency (highest to lowest), and study the top-10 most infected OSs. Due to ethical considerations, we anonymize OS names by renaming according to their actual rank, e.g., top-ranked OS is named os1, 2nd ranked OS is named os2, and so forth. If a lower-ranked OS is a different version of a higher ranked OS, we add the version information to the name, e.g., os1 v2.

Results. We report the results of these experiments on Cerber and Locky in Figures 6 and 7, respectively. In the tables on the right, those OSs that are correctly discovered by RAPPOR, OLH, and Item-CLDP with correct ranks are depicted in bold. OSs that are correctly discovered but have incorrect rank are depicted in regular font. OSs that privacy methods claim to be among top-
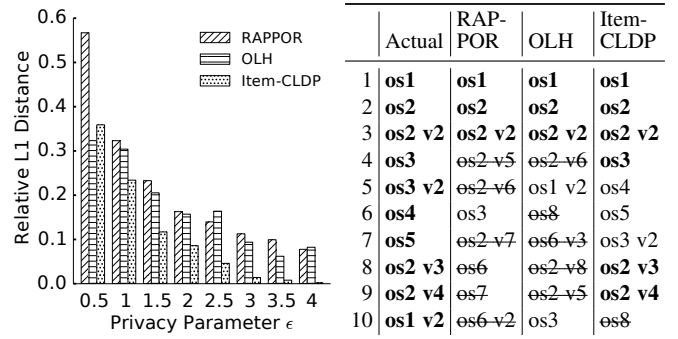

Fig. 6: Analyzing OS vulnerability for ransomware Cerber. *Left*: L1 distances between actual OS counts versus locally private counts reported by RAPPOR, OLH, and our Item-CLDP protocol. *Right*: Actual and locally private top-10 OS rankings when $\varepsilon = 1$. **Bold** = OS with correct rank, regular font = OS with wrong rank, ~~strike-through~~ = OS not in actual top-10. Item-CLDP better preserves relative rankings and generally has lower L1 errors.
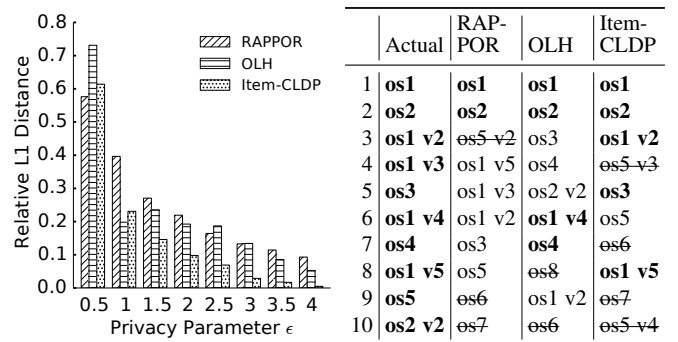

Fig. 7: Analyzing OS vulnerability for ransomware Locky. Refer to Figure 6 for a similar experiment with a different ransomware, same details and conclusions apply to this figure.

10 but in reality are not are depicted with strike-through. We first observe from the tables that the heaviest hitters are correctly discovered in the correct order by all privacy solutions, e.g., top-3 in Cerber. However, as we move lower in the ranking, LDP/CLDP methods start making errors. Particularly for Cerber, RAPPOR and OLH correctly identify only 4 and 5 out of 10 most frequent OSs, respectively, whereas Item-CLDP can identify 9 out of 10. Note that Item-CLDP is missing only the lowest ranked OS (10th), which is arguably the least significant among all ten.

L1 errors in the graphs on the left show that when ε is small (0.5 or 1), LDP is competitive against CLDP in this case study. When ε is higher, CLDP clearly dominates in terms of accuracy. Comparing tabular rankings with L1 scores, we see that CLDP can preserve relative rankings even when its L1 errors are similar to those of LDP. For example, the L1 errors of RAPPOR, OLH, and Item-CLDP are similar when $\varepsilon = 1$. However, studying the top-10 tables shows that Item-CLDP is better at identifying frequent OSs than RAPPOR and OLH.

Case Study #3: Inspecting Suspicious Activity

Setup. In this case study, we consider the sequences of security-related event flags raised by Symantec's behavioral detection engine on each client machine. There are 143 different flags signalling various forms of suspicious activity, ranging from process injection to load point modification. When a flag is raised, it is logged on the client machine with a timestamp, as such the collection of the flagged events constitute a *sequence* over a time period. We investigate the accuracy of collecting these event sequences using Sequence-CLDP. We focus on the same 31-day periods we considered in Case Study #1, and collect locally private event sequences from the same set of machines infected by ransomware Cerber/Locky. Longitudinal analysis of these event sequences enables Symantec to inspect suspicious activities possibly related to the ransomware infection, e.g., chain of anomalous events leading to the infection. This helps in inferring the precursors or consequences of the infection, and Symnatec can update its detection engine based on the findings. In total, we have 23,558 and 5,717 sequences for Cerber and Locky, respectively, with lengths between 2 to 30.

We use n-gram analysis by mining the top-k popular bigram and trigram patterns from the sequences [9], [32]. We mine the actual patterns that would be obtained if no privacy were applied, and the patterns obtained after Sequence-CLDP is applied. Let A denote the set of actual top-k patterns and B denote the set of top-k patterns mined from perturbed data. We measure their similarity using the Jaccard index: $\text{Jaccard}(A, B) = \frac{|A \cap B|}{|A \cup B|}$. Jaccard similarity is between 0 and 1, with values close to 1 indicating higher similarity. We do not compare against RAPPOR, OLH or SVIM in this case study, since they are not compatible with sequence perturbation.

Results. The results are shown in Figure 8. We make two important observations. First, as we relax the privacy requirement by increasing α, n-grams mined from perturbed sequences become more accurate, as implied by the increase in Jaccard similarity. Second, mining fewer top-k patterns is easier than mining many patterns in general. For example, top-20 in Locky has higher Jaccard similarity score than top-30 and top-50. Similar observation applies to Cerber. This shows Sequence-CLDP preserves the heaviest hitters best, and has higher probability of making errors as n-grams become less and less frequent, which agree with our intuition from Case Study #2. Note that we mine more patterns in
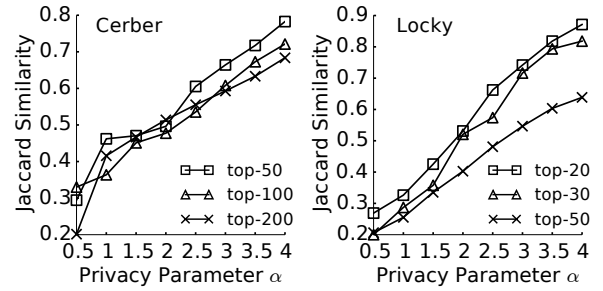

Fig. 8: Utility preservation of Sequence-CLDP in mining top-k n-gram patterns from security event sequences obtained from ransomware infected machines. Sequence-CLDP allows discovery of a high fraction of frequent patterns, which are useful to analyze suspicious activity on the machines.

the case of Cerber (up to top-200 as opposed to top-50 for Locky) since the Cerber dataset has more input sequences, thus we can find more n-grams with significant support and confidence.

5.2 Experiments with Public Datasets

Datasets. We also experimented on two public datasets: POS and Retail. Both are set-valued datasets. We use them to run singleton non-ordinal item experiments as well as set-valued experiments. For the former, we randomly sample an item from each itemset to create a singleton item dataset. For the latter, we run the set-valued adaptation of our Sequence-CLDP protocol and compare it against SVIM, the state-of-the-art set-valued LDP protocol [31].

POS contains several years of market basket sale data from a large electronics retailer [33]. It consists of a total of 515,596 transactions with 1,657 unique items sold.

Retail contains transactions occurring between January 2010 and September 2011 for a UK-based online retail site [34]. After cleaning empty entries, this dataset consists of a total of 540,455 transactions with 2,603 unique items.

Evaluation Metrics. We use the following metrics to evaluate the accuracy of the privacy protocols. Similar to our notation from Section 4.2, let x denote an item, $\text{true}(x)$ denote its true frequency, and $\text{est}(x)$ denote its frequency estimated by the privacy protocol. Let $\mathbf{X}_{gt} = \{x_1, x_2, ..., x_k\}$ be the ground truth top-k items where x_j is the j'th most frequent item.

Average Relative Error (AvRE) measures the mean relative error in top-k items' estimated frequencies versus their true frequencies. Formally:

$$\text{AvRE} = \frac{\sum_{x \in \mathbf{X}_{gt}} \frac{abs(\text{est}(x) - \text{true}(x))}{\text{true}(x)}}{k}$$

The *Kendall-tau coefficient* (KT) measures how well the rankings of heavy hitter top-k items are preserved. A pair of items x, $y \in \mathbf{X}_{gt}$ are said to be concordant if their sorted popularity ranks agree, i.e., either of the following hold:

$$\text{true}(x) > \text{true}(y) \wedge \text{est}(x) > \text{est}(y)$$
$$\text{true}(x) < \text{true}(y) \wedge \text{est}(x) < \text{est}(y)$$

They are said to be discordant if neither holds. Then, the Kendall-tau coefficient of correlation can be defined as:

$$\text{KT} = \frac{(\# \text{ of concordant pairs}) - (\# \text{ of discordant pairs})}{k(k-1)/2}$$

Results of Singleton Experiments. We run experiments in the singleton setting and compare Item-CLDP with LDP protocols
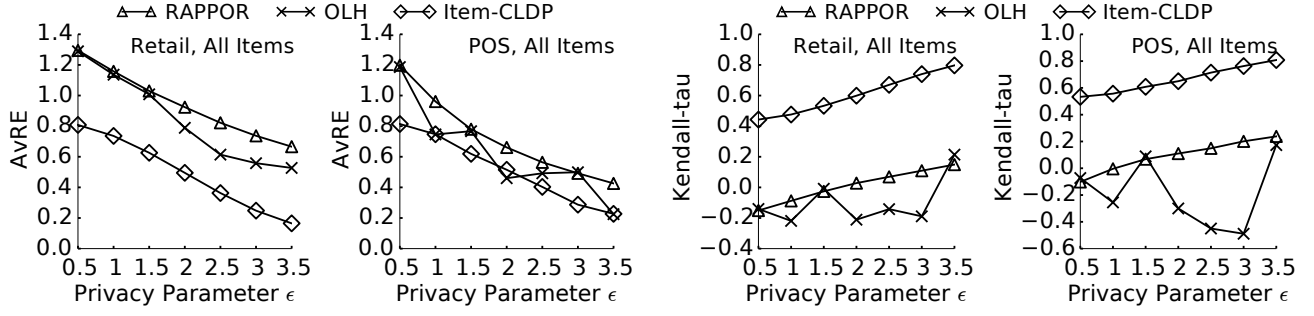
Fig. 9: AvRE (lower is better) and Kendall-tau scores (higher is better) for singleton experiments across all items in datasets Retail and POS. Item-CLDP provides high utility in estimating item frequencies and rankings across a spectrum of privacy parameter settings.
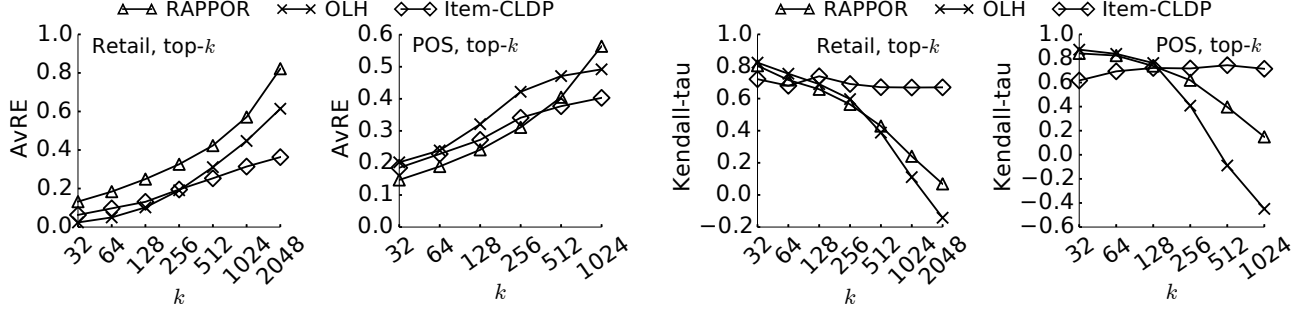

Fig. 10: AvRE and Kendall-tau scores for top-k singletons in Retail and POS. ε fixed to 2.5, varying k on x-axis. Item-CLDP's accuracy is higher than LDP protocols especially for medium-frequency and infrequent items, e.g., $k \geq 256$.

RAPPOR and OLH. Each experiment is repeated 20 times and results are averaged. In Figure 9, we measure AvRE and Kendall-tau across *all* items by setting $k = |\mathcal{U}|$. Results show that as privacy is relaxed (i.e., ε and α increase) AvRE decreases and Kendall-tau increases. In most cases, Item-CLDP provides better accuracy; most noticeably, Kendall-tau scores of Item-CLDP are much higher than those of RAPPOR and OLH. When $\varepsilon \geq 3.5$, Kendall-tau scores indicate almost perfect correlation between actual item rankings and rankings found by Item-CLDP, confirming its high accuracy.

Next, we fix the privacy parameter to $\varepsilon = 2.5$ and vary the k (for top-k) to analyze how the protocols behave with respect to varying popularities of items. The results of this experiment are reported in Figure 10. For small k such as $k \leq 64$, there is usually one LDP protocol at least comparable to or better than Item-CLDP. Note that $k = 64$ is a constrained setting covering less than only 6% of the items in the universe. LDP protocols are optimized to discover such heavy hitters and therefore, they deliver good results when k is small. However, for larger k, we observe that Item-CLDP can significantly outperform RAPPOR and OLH. In particular, for $k \geq 512$, under LDP protocols there is almost no correlation in frequency rankings (implied by Kendall-tau results near or below 0), whereas in Item-CLDP, a strong correlation is maintained across all k. In short, if the goal is to discover only the top few heavy hitters, LDP protocols offer sufficient accuracy. However, if the goal is to find statistics regarding medium-frequency or infrequent items as well, LDP protocols have inadequate accuracy and we recommend using Item-CLDP.

Results of Set-Valued Experiments. We compare the set-valued adaptation of Sequence-CLDP against the SVIM protocol satisfying LDP [31]. Similar to the singleton experiment, we start by setting $k = |\mathcal{U}|$ and vary ε to study the impact of the privacy budget on accuracy across all items. From the results in Figure 11, we observe that Sequence-CLDP offers significant accuracy
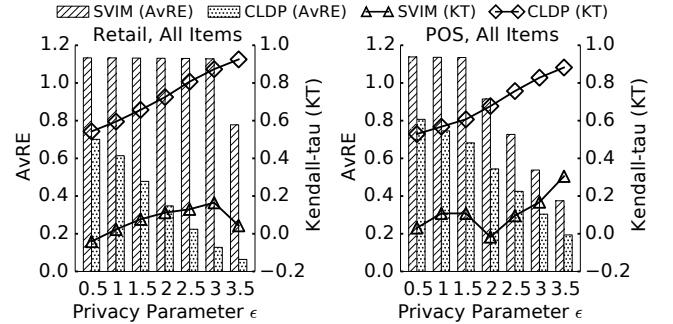

Fig. 11: AvRE (bars, left y-axis) and Kendall-tau scores (lines, right y-axis) for set-valued experiments across all items. Sequence-CLDP preserves item frequencies and rankings more effectively than SVIM.
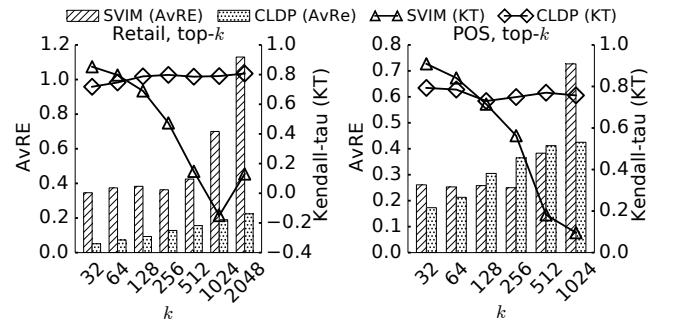

Fig. 12: AvRE (bars, left y-axis) and Kendall-tau scores (lines, right y-axis) for set-valued experiments over top-k items with $\varepsilon = 2.5$. Sequence-CLDP offers higher accuracy in terms of AvRE and Kendall-tau for majority of k values.

improvement in terms of both AvRE and Kendall-tau score. For example, the accuracy improvement in terms of AvRE ranges between 30-90% depending on the dataset and the value of the privacy parameter. In Figure 12, we fix ε to 2.5 and vary k. As we increase k, the accuracy of the protocols generally decrease, since making estimations regarding infrequent items is often more difficult than estimating only the heavy hitters. For Sequence-CLDP, this accuracy decrease is linear or sub-linear; but for SVIM, when $k > 512$, errors start increasing almost exponentially, reiterating that LDP protocols can be poorly suited to estimate statistics regarding infrequent items. Studying the Kendall-tau scores, for very small k, Kendall-tau of SVIM and Sequence-CLDP are similar, whereas when $k > 128$, Sequence-CLDP's Kendall-tau scores are significantly better.

6 RELATED WORK

Differential privacy was initially proposed in the centralized setting in which a trusted central data collector possesses a database containing clients' true values, and noise is applied on the database or queries executed on the database instead of each client's individual value [18], [35]. In contrast, in LDP, each client locally perturbs their data on their device before sending the perturbed version to the data collector [1]. The local setting has seen practical real-world deployment, including Google's RAPPOR as a Chrome extension [2], [4], Apple's use of LDP for spelling prediction and emoji frequency detection [5], [6], and Microsoft's collection of application telemetry [7].

LDP has also sparked interest from the academic community. There have been several theoretical treatments for finding upper and lower bounds on the accuracy and utility of LDP [1], [19], [36], [37], [38]. From a more practical perspective, Wang et al. [3] showed the optimality of OLH for singleton item frequency estimation. Qin et al. [21] and Wang et al. [31] studied frequent item and itemset mining from set-valued client data. Cormode et al. [39] and Zhang et al. [40] studied the problem of obtaining marginal tables from high-dimensional data. Recently, LDP was considered in the contexts of geolocations [41], decentralized social graphs [42], and discovering emerging terms from text [43].

However, there have also been criticisms and concerns regarding the utility of LDP, which motivated recent works proposing relaxations or alternatives to LDP. BLENDER [44] proposed a hybrid privacy model in which only a subset of users enjoy LDP, whereas remaining users act as opt-in beta testers who receive the guarantees of centralized DP. In contrast, our work stays purely in the local privacy model without requiring a trusted data collector (necessary in centralized or hybrid DP) or opt-in clients. Personalized LDP, a weaker form of LDP, was proposed for spatial data aggregation in [41]; whereas the Restricted LDP scheme proposed in [45] treats certain client data as more sensitive than others and suggests restricted perturbation schemes to specifically address the more sensitive data. In contrast, our CLDP approach treats all users' data as sensitive (parallel with LDP assumptions) and remains agnostic and extensible with respect to data types.

7 CONCLUSION

In this paper, we proposed *Condensed Local Differential Privacy (CLDP)* for utility-aware and privacy-preserving data collection, and developed three protocols: Ordinal-CLDP, Item-CLDP, and Sequence-CLDP for handling a variety of data types prevalent in the cybersecurity domain. Our protocols remain accurate for populations that are an order of magnitude smaller than those required by existing LDP protocols to give adequate accuracy. Case studies using Symantec datasets and experiments on public datasets demonstrate the utility of our proposed approach.

REFERENCES

[1] J. C. Duchi, M. I. Jordan, and M. J. Wainwright, "Local privacy and statistical minimax rates," in *2013 IEEE 54th Annual Symposium on Foundations of Computer Science (FOCS)*. IEEE, 2013, pp. 429–438.

[2] Ú. Erlingsson, V. Pihur, and A. Korolova, "Rappor: Randomized aggregatable privacy-preserving ordinal response," in *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 2014, pp. 1054–1067.

[3] T. Wang, J. Blocki, N. Li, and S. Jha, "Locally differentially private protocols for frequency estimation," in *Proc. of the 26th USENIX Security Symposium*, 2017, pp. 729–745.

[4] G. Fanti, V. Pihur, and Ú. Erlingsson, "Building a rappor with the unknown: Privacy-preserving learning of associations and data dictionaries," *Proceedings on Privacy Enhancing Technologies*, vol. 2016, no. 3, pp. 41–61, 2016.

[5] A. G. Thakurta, A. H. Vyrros, U. S. Vaishampayan, G. Kapoor, J. Freudiger, V. R. Sridhar, and D. Davidson, "Learning new words," Mar. 14 2017, uS Patent 9,594,741.

[6] A. G. Thakurta, A. H. Vyrros, U. S. Vaishampayan, G. Kapoor, J. Freudiger, V. V. Prakash, A. Legendre, and S. Duplinsky, "Emoji frequency detection and deep link frequency," Dec. 14 2017, uS Patent App. 15/640,266.

[7] B. Ding, J. Kulkarni, and S. Yekhanin, "Collecting telemetry data privately," in *Advances in Neural Information Processing Systems*, 2017, pp. 3571–3580.

[8] D. H. Chau, C. Nachenberg, J. Wilhelm, A. Wright, and C. Faloutsos, "Polonium: Tera-scale graph mining and inference for malware detection," in *Proceedings of the 2011 SIAM International Conference on Data Mining*. SIAM, 2011, pp. 131–142.

[9] N. Karampatziakis, J. W. Stokes, A. Thomas, and M. Marinescu, "Using file relationships in malware classification," in *International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment*. Springer, 2012, pp. 1–20.

[10] A. Tamersoy, K. Roundy, and D. H. Chau, "Guilt by association: large scale malware detection by mining file-relation graphs," in *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2014, pp. 1524–1533.

[11] M. Egele, T. Scholte, E. Kirda, and C. Kruegel, "A survey on automated dynamic malware-analysis techniques and tools," *ACM Computing Surveys (CSUR)*, vol. 44, no. 2, p. 6, 2012.

[12] F. McSherry and K. Talwar, "Mechanism design via differential privacy," in *48th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*. IEEE, 2007, pp. 94–103.

[13] N. Li, W. Qardaji, D. Su, Y. Wu, and W. Yang, "Membership privacy: a unifying framework for privacy definitions," in *Proceedings of the 2013 ACM SIGSAC conference on Computer & Communications Security*. ACM, 2013, pp. 889–900.

[14] B. Yang, I. Sato, and H. Nakagawa, "Bayesian differential privacy on correlated data," in *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data*. ACM, 2015, pp. 747–762.

[15] S. P. Kasiviswanathan and A. Smith, "On the semantics of differential privacy: A bayesian formulation," *Journal of Privacy and Confidentiality*, vol. 6, no. 1, 2014.

[16] L. Yu, L. Liu, and C. Pu, "Dynamic differential location privacy with personalized error bounds," in *NDSS*, 2017.

[17] S. L. Warner, "Randomized response: A survey technique for eliminating evasive answer bias," *Journal of the American Statistical Association*, vol. 60, no. 309, pp. 63–69, 1965.

[18] A. Inan, M. E. Gursoy, and Y. Saygin, "Sensitivity analysis for non-interactive differential privacy: bounds and efficient algorithms," *IEEE Transactions on Dependable and Secure Computing*, 2017.

[19] R. Bassily and A. Smith, "Local, private, efficient protocols for succinct histograms," in *Proceedings of the 47th Annual ACM Symposium on Theory of Computing*. ACM, 2015, pp. 127–135.

[20] R. Bassily, U. Stemmer, A. G. Thakurta *et al.*, "Practical locally private heavy hitters," in *Advances in Neural Information Processing Systems*, 2017, pp. 2288–2296.

[21] Z. Qin, Y. Yang, T. Yu, I. Khalil, X. Xiao, and K. Ren, "Heavy hitter estimation over set-valued data with local differential privacy," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 2016, pp. 192–203.

[22] K. Chatzikokolakis, M. E. Andres, N. E. Bordenabe, and C. Palamidessi, "Broadening the scope of differential privacy using metrics," in *International Symposium on Privacy Enhancing Technologies (PETS)*. Springer, 2013, pp. 82–102.

[23] M. Andres, N. Bordenabe, K. Chatzikokolakis, and C. Palamidessi, "Geo-indistinguishability: Differential privacy for location-based systems," in *Proceedings of the 2013 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 2013, pp. 901–914.

[24] N. Bordenabe, K. Chatzikokolakis, and C. Palamidessi, "Optimal geo-indistinguishable mechanisms for location privacy," in *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 2014, pp. 251–262.

[25] M. Hay, C. Li, G. Miklau, and D. Jensen, "Accurate estimation of the degree distribution of private networks," in *9th IEEE International Conference on Data Mining (ICDM)*. IEEE, 2009, pp. 169–178.

[26] Z. Ding, Y. Wang, G. Wang, D. Zhang, and D. Kifer, "Detecting violations of differential privacy," in *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 2018, pp. 475–489.

[27] P. Kairouz, S. Oh, and P. Viswanath, "The composition theorem for differential privacy," *IEEE Transactions on Information Theory*, vol. 63, no. 6, pp. 4037–4049, 2017.

[28] C. Liu, X. He, T. Chanyaswad, S. Wang, and P. Mittal, "Investigating statistical privacy frameworks from the perspective of hypothesis testing," *Proceedings on Privacy Enhancing Technologies*, vol. 2019, no. 3, pp. 233–254, 2019.

[29] J. Soria-Comas, J. Domingo-Ferrer, D. Sánchez, and S. Martínez, "Enhancing data utility in differential privacy via microaggregation-based k-anonymity," *The VLDB Journal – The International Journal on Very Large Data Bases*, vol. 23, no. 5, pp. 771–794, 2014.

[30] D. Sánchez, M. Batet, D. Isern, and A. Valls, "Ontology-based semantic similarity: A new feature-based approach," *Expert Systems with Applications*, vol. 39, no. 9, pp. 7718–7728, 2012.

[31] T. Wang, N. Li, and S. Jha, "Locally differentially private frequent itemset mining," in *IEEE Symposium on Security and Privacy (SP)*. IEEE, 2018.

[32] S. B. Mehdi, A. K. Tanwani, and M. Farooq, "Imad: in-execution malware analysis and detection," in *Proceedings of the 11th Annual Conference on Genetic and Evolutionary Computation*. ACM, 2009, pp. 1553–1560.

[33] "Pos dataset," https://github.com/cpearce/HARM/blob/master/datasets/BMS-POS.csv, 2015, [Online; accessed 05-Oct-2018].

[34] D. Chen, S. L. Sain, and K. Guo, "Data mining for the online retail industry: A case study of rfm model-based customer segmentation using data mining," *Journal of Database Marketing & Customer Strategy Management*, vol. 19, no. 3, pp. 197–208, 2012.

[35] C. Dwork, "Differential privacy," in *International Colloquium on Automata, Languages, and Programming*. Springer, 2006, pp. 1–12.

[36] P. Kairouz, S. Oh, and P. Viswanath, "Extremal mechanisms for local differential privacy," in *Advances in Neural Information Processing Systems*, 2014, pp. 2879–2887.

[37] A. Smith, A. Thakurta, and J. Upadhyay, "Is interaction necessary for distributed private learning?" in *2017 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2017, pp. 58–77.

[38] M. Bun, J. Nelson, and U. Stemmer, "Heavy hitters and the structure of local privacy," in *Proceedings of the 35th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems*. ACM, 2018, pp. 435–447.

[39] G. Cormode, T. Kulkarni, and D. Srivastava, "Marginal release under local differential privacy," in *Proceedings of the 2018 International Conference on Management of Data*. ACM, 2018, pp. 131–146.

[40] Z. Zhang, T. Wang, N. Li, S. He, and J. Chen, "Calm: Consistent adaptive local marginal for marginal release under local differential privacy," in *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 2018, pp. 212–229.

[41] R. Chen, H. Li, A. Qin, S. P. Kasiviswanathan, and H. Jin, "Private spatial data aggregation in the local setting," in *2016 IEEE 32nd International Conference on Data Engineering (ICDE)*. IEEE, 2016, pp. 289–300.

[42] Z. Qin, T. Yu, Y. Yang, I. Khalil, X. Xiao, and K. Ren, "Generating synthetic decentralized social graphs with local differential privacy," in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 2017, pp. 425–438.

[43] N. Wang, X. Xiao, Y. Yang, T. D. Hoang, H. Shin, J. Shin, and G. Yu, "Privtrie: Effective frequent term discovery under local differential privacy," in *IEEE International Conference on Data Engineering (ICDE)*, 2018.

[44] B. Avent, A. Korolova, D. Zeber, T. Hovden, and B. Livshits, "BLENDER: Enabling local search with a hybrid differential privacy model," in *26th USENIX Security Symposium (USENIX Security 17)*. Vancouver, BC: USENIX Association, 2017, pp. 747–764.

[45] T. Murakami and Y. Kawamoto, "Restricted local differential privacy for distribution estimation with high data utility," *arXiv preprint arXiv:1807.11317*, 2018.

Mehmet Emre Gursoy is a PhD student in the School of Computer Science at Georgia Institute of Technology. He received his MS degree in Computer Science from University of California Los Angeles (UCLA) and his BS degree in Computer Science and Engineering from Sabanci University, Turkey. His research interests include data privacy, security, machine learning, mobile computing, and big data analytics.

Acar Tamersoy is a Senior Research Scientist at Symantec Research Labs. He received his PhD in Computer Science from Georgia Institute of Technology, his MS in Biomedical Informatics from Vanderbilt University, and his BS in Computer Science and Engineering from Sabanci University, Turkey. His research interests include data mining, with a focus on security, privacy, healthcare, and finance problems.

Stacey Truex is a PhD student in the School of Computer Science at Georgia Institute of Technology. She received her MS degree from the University of Washington in Computer Science and Systems and her BS in Computer Science and BA in Mathematics from Wake Forest University. Her research interests include data privacy, security, federated learning systems, and big data analytics.

Wenqi Wei is currently pursuing his PhD in the School of Computer Science, Georgia Institute of Technology. He received his BE degree from the School of Electronic Information and Communications, Huazhong University of Science and Technology. His research interests include data privacy, security, machine learning, and big data analytics.

Ling Liu is a full Professor in the College of Computing at Georgia Institute of Technology. She is an elected IEEE Fellow, a recipient of the IEEE Computer Society Technical Achievement award in 2012, and a recipient of the best paper award from a dozen of top venues including ICDCS 2003, WWW 2004, 2005 Pat Goldberg Memorial Best Paper Award, IEEE Cloud 2012, IEEE ICWS 2013, Mobiquitous 2014, APWeb 2015, IEEE/ACM CCGrid 2015, IEEE Symposium on Big Data 2016, IEEE Edge 2017 and IEEE IoT 2017. She served as the general chair and PC chair of numerous IEEE and ACM conferences, as well as on the editorial boards of over a dozen international journals. Her current research is primarily sponsored by NSF, IBM, and Intel.

Supplementary Material for "Secure and Utility-Aware Data Collection with Condensed Local Differential Privacy"

Mehmet Emre Gursoy, *Student Member, IEEE,* Acar Tamersoy, Stacey Truex, Wenqi Wei, and Ling Liu, *Fellow, IEEE*

---❖---

1 PROOF OF THEOREM 1

Recall that for item universe $\mathcal{U}$, Exponential Mechanism (EM), denoted by Φ_{EM}, takes as input true value v and produces fake value $y \in \mathcal{U}$ with probability:

$$\Pr[\Phi_{EM}(v) = y] = \frac{e^{\frac{-\alpha \cdot d(v,y)}{2}}}{\sum_{z \in \mathcal{U}} e^{\frac{-\alpha \cdot d(v,z)}{2}}}$$

We prove here that Φ_{EM} satisfies α-CLDP by showing that:

$$\frac{\Pr[\Phi_{EM}(v_1) = y]}{\Pr[\Phi_{EM}(v_2) = y]} \le e^{\alpha \cdot d(v_1, v_2)}$$

Proof. We start by applying the definition of EM and breaking the odds ratio into two terms:

$$\frac{\Pr[\Phi_{EM}(v_1) = y]}{\Pr[\Phi_{EM}(v_2) = y]} = \frac{\frac{e^{\frac{-\alpha \cdot d(v_1,y)}{2}}}{\sum_{z \in \mathcal{U}} e^{\frac{-\alpha \cdot d(v_1,z)}{2}}}}{\frac{e^{\frac{-\alpha \cdot d(v_2,y)}{2}}}{\sum_{z \in \mathcal{U}} e^{\frac{-\alpha \cdot d(v_2,z)}{2}}}} \tag{1}$$

$$= \underbrace{\frac{e^{\frac{-\alpha \cdot d(v_1,y)}{2}}}{e^{\frac{-\alpha \cdot d(v_2,y)}{2}}}}_{*} \cdot \underbrace{\frac{\sum_{z \in \mathcal{U}} e^{\frac{-\alpha \cdot d(v_2,z)}{2}}}{\sum_{z \in \mathcal{U}} e^{\frac{-\alpha \cdot d(v_1,z)}{2}}}}_{**} \tag{2}$$

For *, we observe that:

$$\frac{e^{\frac{-\alpha \cdot d(v_1,y)}{2}}}{e^{\frac{-\alpha \cdot d(v_2,y)}{2}}} = e^{\frac{\alpha \cdot \left(d(v_2,y) - d(v_1,y)\right)}{2}} \tag{3}$$

Since d is a metric, it satisfies the triangle inequality. Therefore, it holds that: $d(v_2, y) - d(v_1, y) \le d(v_1, v_2)$. Combining this with the above, we conclude for *:

$$\frac{e^{\frac{-\alpha \cdot d(v_1,y)}{2}}}{e^{\frac{-\alpha \cdot d(v_2,y)}{2}}} \le e^{\frac{\alpha \cdot d(v_1,v_2)}{2}} \tag{4}$$

Next, we study the second term **:

$$\frac{\sum_{z \in \mathcal{U}} e^{\frac{-\alpha \cdot d(v_2,z)}{2}}}{\sum_{z \in \mathcal{U}} e^{\frac{-\alpha \cdot d(v_1,z)}{2}}} = \frac{\sum_{z \in \mathcal{U}} e^{\frac{-\alpha \cdot d(v_2,z) + \alpha \cdot d(v_1,z) - \alpha \cdot d(v_1,z)}{2}}}{\sum_{z \in \mathcal{U}} e^{\frac{-\alpha \cdot d(v_1,z)}{2}}} \tag{5}$$

Again by triangle inequality: $d(v_1, z) - d(v_2, z) \le d(v_1, v_2)$. Applying this to the numerator we get:

$$\frac{\sum_{z \in \mathcal{U}} e^{\frac{-\alpha \cdot d(v_2,z)}{2}}}{\sum_{z \in \mathcal{U}} e^{\frac{-\alpha \cdot d(v_1,z)}{2}}} \le \frac{\sum_{z \in \mathcal{U}} e^{\frac{\alpha \cdot d(v_1,v_2) - \alpha \cdot d(v_1,z)}{2}}}{\sum_{z \in \mathcal{U}} e^{\frac{-\alpha \cdot d(v_1,z)}{2}}} \tag{6}$$

$$\le \frac{e^{\frac{\alpha \cdot d(v_1,v_2)}{2}} \cdot \sum_{z \in \mathcal{U}} e^{\frac{-\alpha \cdot d(v_1,z)}{2}}}{\sum_{z \in \mathcal{U}} e^{\frac{-\alpha \cdot d(v_1,z)}{2}}} \tag{7}$$

$$\le e^{\frac{\alpha \cdot d(v_1,v_2)}{2}} \tag{8}$$

We established that $* \le e^{\frac{\alpha \cdot d(v_1,v_2)}{2}}$ and $** \le e^{\frac{\alpha \cdot d(v_1,v_2)}{2}}$. Plugging them into Equation 2 concludes our proof:

$$\frac{\Pr[\Phi_{EM}(v_1) = y]}{\Pr[\Phi_{EM}(v_2) = y]} \le e^{\frac{\alpha \cdot d(v_1,v_2)}{2}} \cdot e^{\frac{\alpha \cdot d(v_1,v_2)}{2}} = e^{\alpha \cdot d(v_1,v_2)}$$

$\square$

2 UTILITY PROPERTIES OF ORDINAL-CLDP

In Section 4.1 of the main text, we claimed that when d is selected with the property that $d(v_i, v_j) = c$ for $v_i \ne v_j$, where c is a constant, then the aggregation of the outputs of Algorithm 1 in the main text (denoted Φ_{ORD}) preserves the relative frequency rank of v_i and v_j in expectation. Note that since d is a metric, by definition, $d(v_i, v_i) = 0$ for all $v_i \in \mathcal{U}$.

In order to prove this claim, we first introduce some notation. Let $v_i \in \mathcal{U}$ be an item, let $\text{true}(v_i)$ be the true count of v_i in the population, and let $\text{obs}(v_i)$ be the observed count of v_i from the aggregation of perturbed outputs of Algorithm 1. Then, the above claim is equivalent to the statement that iff $\text{true}(v_i) > \text{true}(v_j)$, then $\mathbb{E}[\text{obs}(v_i)] > \mathbb{E}[\text{obs}(v_j)]$. Given this holds for all $v_i, v_j \in \mathcal{U}$, the converse follows trivially that iff $\text{true}(v_i) < \text{true}(v_j)$, then $\mathbb{E}[\text{obs}(v_i)] < \mathbb{E}[\text{obs}(v_j)]$. Also, if the relative frequency rank relation between all pairs of individual items are preserved in expectation, it implies that that the complete ranking is also preserved. Thus, it suffices to prove the initial statement.

We begin the proof by expressing $\mathbb{E}[\text{obs}(v_i)]$:

$$\mathbb{E}[\text{obs}(v_i)] = \text{true}(v_i) \cdot \Pr[\Phi_{ORD}(v_i) = v_i]$$
$$+ \text{true}(v_j) \cdot \Pr[\Phi_{ORD}(v_j) = v_i]$$
$$+ \sum_{v_k \in \mathcal{U} \setminus \{v_i, v_j\}} \text{true}(v_k) \cdot \Pr[\Phi_{ORD}(v_k) = v_i]$$

Similarly, $\mathbb{E}[\mathrm{obs}(v_j)]$ can be expressed as:

$$\begin{aligned}
\mathbb{E}[\mathrm{obs}(v_j)] = \ & \mathrm{true}(v_j) \cdot \Pr[\Phi_{ORD}(v_j) = v_j] \\
& + \mathrm{true}(v_i) \cdot \Pr[\Phi_{ORD}(v_i) = v_j] \\
& + \sum_{v_k \in \mathcal{U} \setminus \{v_i, v_j\}} \mathrm{true}(v_k) \cdot \Pr[\Phi_{ORD}(v_k) = v_j]
\end{aligned}$$

For the above definition of d, we observe by construction of Φ_{ORD} in the main text that $\Pr[\Phi_{ORD}(v_i) = v_i] = \Pr[\Phi_{ORD}(v_j) = v_j]$, $\Pr[\Phi_{ORD}(v_j) = v_i] = \Pr[\Phi_{ORD}(v_i) = v_j]$, and $\Pr[\Phi_{ORD}(v_k) = v_i] = \Pr[\Phi_{ORD}(v_k) = v_j]$. Let constants c_1, c_2, and c_3 denote these three quantities respectively. Then, $\mathbb{E}[\mathrm{obs}(v_i)] - \mathbb{E}[\mathrm{obs}(v_j)]$ can be written as:

$$\begin{aligned}
\mathbb{E}[\mathrm{obs}(v_i)] - \mathbb{E}[\mathrm{obs}(v_j)] = \ & c_1 \cdot (\mathrm{true}(v_i) - \mathrm{true}(v_j)) \\
& - c_2 \cdot (\mathrm{true}(v_i) - \mathrm{true}(v_j)) + c_3 \cdot \sum_{v_k \in \mathcal{U} \setminus \{v_i, v_j\}} (\mathrm{true}(v_k) - \mathrm{true}(v_k))
\end{aligned}$$

which can be simplified to:

$$\mathbb{E}[\mathrm{obs}(v_i)] - \mathbb{E}[\mathrm{obs}(v_j)] = (c_1 - c_2) \cdot (\mathrm{true}(v_i) - \mathrm{true}(v_j)) \quad (9)$$

By construction of Φ_{ORD}, we know that $c_1 > c_2$. Then, Equation 9 shows that $\mathbb{E}[\mathrm{obs}(v_i)] - \mathbb{E}[\mathrm{obs}(v_j)]$ and $\mathrm{true}(v_i) - \mathrm{true}(v_j)$ are directly positively correlated. In other words, iff $\mathrm{true}(v_i) > \mathrm{true}(v_j)$, then $\mathbb{E}[\mathrm{obs}(v_i)] > \mathbb{E}[\mathrm{obs}(v_j)]$ and vice versa. This concludes the proof.

3 COMPOSITION IN ITEM-CLDP

The Item-CLDP protocol from the main text is a two-round protocol, where the first round satisfies $(\alpha \cdot L)$-CLDP with d, and the second round satisfies $\alpha \cdot (1 - L)$-CLDP with d'. We argued that the composition of the two rounds satisfies:

$$\begin{aligned}
\frac{\Pr[\Phi_{ITEM}(v_1) = \langle v', v'' \rangle]}{\Pr[\Phi_{ITEM}(v_2) = \langle v', v'' \rangle]} & \leq e^{\alpha \cdot L \cdot d(v_1, v_2)} \cdot e^{\alpha \cdot (1-L) \cdot d'(v_1, v_2)} \\
& \leq e^{\alpha \cdot \max\{d(v_1, v_2), d'(v_1, v_2)\}}
\end{aligned}$$

which is enabled by the fact that $d^*(v_1, v_2) = \max\{d(v_1, v_2), d'(v_1, v_2)\}$ is also a metric, given that d and d' are metrics by their construction within Item-CLDP.

Here, we prove this fact that enables Item-CLDP composition. That is, given $d(v_i, v_j)$ and $d'(v_i, v_j)$ are metrics, $d^*(v_i, v_j) = \max\{d(v_i, v_j), d'(v_i, v_j)\}$ is also a metric.

First recall the properties of being a metric, which are satisfied by d and d'. We write these properties for d, as d' is analogous:

1) $d(v_i, v_j) \geq 0, \forall v_i, v_j \in \mathcal{U}$
2) $d(v_i, v_j) = 0$ if and only if $v_i = v_j$
3) $d(v_i, v_j) = d(v_j, v_i), \forall v_i, v_j \in \mathcal{U}$
4) $d(v_i, v_k) \leq d(v_i, v_j) + d(v_j, v_k), \forall v_i, v_j, v_k \in \mathcal{U}$

To prove that d^* is also a metric, we must show that d^* satisfies the four properties above. We prove each property one by one.

Property 1: We know already that $d(v_i, v_j) \geq 0$ and $d'(v_i, v_j) \geq 0$ for all v_i, v_j. $d^*(v_i, v_j)$ takes the maximum of these two non-negative values, which must also be non-negative. Thus, $d^*(v_i, v_j) \geq 0$.

Property 2: There are two directions to prove. First, when $v_i = v_j$ then $d(v_i, v_j) = 0$ and $d'(v_i, v_j) = 0$. As a result, $d^*(v_i, v_j) = \max\{0, 0\} = 0$, which concludes the first direction.

For the second direction, we prove by contradiction. Assume that $d^*(v_i, v_j) = 0$, but contrary to expectation $v_i \neq v_j$. If $v_i \neq v_j$, by the first and second properties satisfied by d and d', either $d(v_i, v_j) > 0$ or $d'(v_i, v_j) > 0$ must hold. However, if either

holds, the maximum of $d(v_i, v_j)$ and $d'(v_i, v_j)$ cannot be zero, since at least one of them is positive. Thus, the initial assumption of $d^*(v_i, v_j) = 0$ is violated, which yields a contradiction. It must be that $v_i = v_j$.

Property 3: The symmetry property of d^* follows from the individual symmetry properties satisfied by d and d'.

$$\begin{aligned}
d^*(v_i, v_j) & = \max\{d(v_i, v_j), d'(v_i, v_j)\} \\
& = \max\{d(v_j, v_i), d'(v_j, v_i)\} \\
& = d^*(v_j, v_i)
\end{aligned}$$

Property 4: By definition of d^*, $d^*(v_i, v_k)$ is either equal to $d(v_i, v_k)$ or $d'(v_i, v_k)$. Without loss of generality assume $d^*(v_i, v_k) = d(v_i, v_k)$; the case with $d^*(v_i, v_k) = d'(v_i, v_k)$ is identical. Using the fourth property of d, we can derive:

$$\begin{aligned}
d^*(v_i, v_k) & = d(v_i, v_k) \\
& \leq d(v_i, v_j) + d(v_j, v_k) \\
& \leq d^*(v_i, v_j) + d^*(v_j, v_k)
\end{aligned}$$

since for each individual term on the right hand side, it is guaranteed by definition of d^* that $d^*(v_i, v_j) \geq d(v_i, v_j)$ and $d^*(v_j, v_k) \geq d(v_j, v_k)$.

4 PROOF OF THEOREM 2

This section contains the privacy proofs for the Sequence-CLDP perturbation mechanism given in Algorithm 2 of the main text. We have claimed in the main text that Sequence-CLDP satisfies two properties: length indistinguishability and content indistinguishability. We organize this section into two subsections for proving each property.

4.1 Length Indistinguishability

We first prove that Sequence-CLDP, denoted here onwards by Φ_{SEQ}, satisfies α-length-indistinguishability when certain value ranges are enforced for its parameters *halt*, *gen*. We need to show that for any pair of true sequences X, Y:

$$\frac{\Pr[\Phi_{SEQ}(X) \rightsquigarrow \ell]}{\Pr[\Phi_{SEQ}(Y) \rightsquigarrow \ell]} \leq e^{\alpha \cdot abs(|X| - |Y|)}$$

Proof. Observe from Algorithm 2 in the main text that given a true sequence X of length $|X|$, Φ_{SEQ} produces an output sequence of length ℓ with probability:

$$\Pr[\Phi_{SEQ}(X) \rightsquigarrow \ell] = \begin{cases} halt \cdot (1 - halt)^\ell \\ \qquad\qquad \text{when } \ell < |X| \\[2ex] (1 - halt)^{|X|} \cdot (1 - gen) \cdot gen^{\ell - |X|} \\ \qquad\qquad \text{when } \ell \geq |X| \end{cases}$$

We consider several disjoint cases depending on the length of true sequences X, Y. The parameters must be chosen so that all cases are simultaneously satisfied.

Case 0: $|X| = |Y|$. This case is trivial since Φ_{SEQ} treats their length equally, resulting in:

$$\frac{\Pr[\Phi_{SEQ}(X) \rightsquigarrow \ell]}{\Pr[\Phi_{SEQ}(Y) \rightsquigarrow \ell]} = 1 = e^{\alpha \cdot 0} \qquad \text{if } |X| = |Y|$$

Remaining cases fall under $|X| \neq |Y|$, and are analyzed case-by-case below.

Case 1: $\ell < |X|$ and $\ell < |Y|$

$$\frac{\Pr[\Phi_{SEQ}(X) \rightsquigarrow \ell]}{\Pr[\Phi_{SEQ}(Y) \rightsquigarrow \ell]} = \frac{halt \cdot (1 - halt)^\ell}{halt \cdot (1 - halt)^\ell} = 1 \le e^{\alpha \cdot abs(|X| - |Y|)}$$

Since $abs(|X| - |Y|) \ge 0$, we trivially have $e^{\alpha \cdot abs(|X| - |Y|)} \ge 1$, hence this case is satisfied without placing constraints on the values of $halt$, gen.

Case 2: $\ell \ge |X|$ and $\ell \ge |Y|$

$$\frac{\Pr[\Phi_{SEQ}(X) \rightsquigarrow \ell]}{\Pr[\Phi_{SEQ}(Y) \rightsquigarrow \ell]} = \frac{(1 - halt)^{|X|} \cdot (1 - gen) \cdot gen^{\ell - |X|}}{(1 - halt)^{|Y|} \cdot (1 - gen) \cdot gen^{\ell - |Y|}}$$
$$= (1 - halt)^{|X| - |Y|} \cdot gen^{|Y| - |X|}$$

Divide this into two subcases:

2a: $|X| < |Y|$: When this holds, the requirement that

$$(1 - halt)^{|X| - |Y|} \cdot gen^{|Y| - |X|} \le e^{\alpha \cdot abs(|X| - |Y|)}$$

can be written as:

$$\frac{gen^{|Y| - |X|}}{(1 - halt)^{|Y| - |X|}} \le e^{\alpha \cdot (|Y| - |X|)}$$
$$(\frac{gen}{1 - halt})^{|Y| - |X|} \le e^{\alpha \cdot (|Y| - |X|)}$$

Thus, we have the constraints for parameters $halt$, gen as:

$$\frac{gen}{1 - halt} \le e^\alpha \tag{10}$$

2b: $|X| > |Y|$: When this holds, the same requirement can be written as:

$$\frac{(1 - halt)^{|X| - |Y|}}{gen^{|X| - |Y|}} \le e^{\alpha \cdot (|X| - |Y|)}$$
$$(\frac{1 - halt}{gen})^{|X| - |Y|} \le e^{\alpha \cdot (|X| - |Y|)}$$

resulting in the parameter constraints:

$$\frac{1 - halt}{gen} \le e^\alpha \tag{11}$$

Case 3: $\ell < |X|$ and $\ell \ge |Y|$

$$\frac{\Pr[\Phi_{SEQ}(X) \rightsquigarrow \ell]}{\Pr[\Phi_{SEQ}(Y) \rightsquigarrow \ell]} \le e^{\alpha \cdot abs(|X| - |Y|)}$$
$$\frac{halt \cdot (1 - halt)^\ell}{(1 - halt)^{|Y|} \cdot (1 - gen) \cdot gen^{\ell - |Y|}} \le e^{\alpha \cdot abs(|X| - |Y|)}$$
$$\frac{halt}{1 - gen} \cdot (\frac{1 - halt}{gen})^{\ell - |Y|} \le e^{\alpha \cdot abs(|X| - |Y|)}$$

Since the assumption in this case is $|Y| \le \ell < |X|$, we can rewrite the RHS as:

$$\frac{halt}{1 - gen} \cdot (\frac{1 - halt}{gen})^{\ell - |Y|} \le e^{\alpha \cdot (\ell - |Y|)} \cdot e^{\alpha \cdot (|X| - \ell)}$$

Given the constraint we established in Equation 11 holds, the following is a sufficient condition to satisfy the above:

$$\frac{halt}{1 - gen} \le e^{\alpha \cdot (|X| - \ell)}$$

Notice that, by assumption, $|X| > \ell$ and both $|X|$ and ℓ are integers representing sequence length. Then, $|X| - \ell \ge 1$, making the following parameter constraint sufficient:

$$\frac{halt}{1 - gen} \le e^\alpha \tag{12}$$

Case 4: $\ell \ge |X|$ and $\ell < |Y|$

$$\frac{\Pr[\Phi_{SEQ}(X) \rightsquigarrow \ell]}{\Pr[\Phi_{SEQ}(Y) \rightsquigarrow \ell]} \le e^{\alpha \cdot abs(|X| - |Y|)}$$
$$\frac{(1 - halt)^{|X|} \cdot (1 - gen) \cdot gen^{\ell - |X|}}{halt \cdot (1 - halt)^\ell} \le e^{\alpha \cdot abs(|X| - |Y|)}$$
$$\frac{1 - gen}{halt} \cdot (\frac{gen}{1 - halt})^{\ell - |X|} \le e^{\alpha \cdot abs(|X| - |Y|)}$$

Since the assumption in this case is $|X| \le \ell < |Y|$, we can rewrite the RHS as:

$$\frac{1 - gen}{halt} \cdot (\frac{gen}{1 - halt})^{\ell - |X|} \le e^{\alpha \cdot (|Y| - \ell)} \cdot e^{\alpha \cdot (\ell - |X|)}$$

Given the constraint we established in Equation 10 holds, the following is a sufficient condition to satisfy the above:

$$\frac{1 - gen}{halt} \le e^{\alpha \cdot (|Y| - \ell)}$$

Since $|Y| > \ell$ and both $|Y|$ and ℓ are integers, we have $|Y| - \ell \ge 1$, making the following parameter constraint sufficient:

$$\frac{1 - gen}{halt} \le e^\alpha \tag{13}$$

Combine all cases: Finally, we combine the parameter constraints we identified at the end of each case (Equations 10, 11, 12, and 13) to obtain a system of equations:

$$\frac{gen}{1 - halt} \le e^\alpha \qquad \frac{1 - halt}{gen} \le e^\alpha$$
$$\frac{halt}{1 - gen} \le e^\alpha \qquad \frac{1 - gen}{halt} \le e^\alpha$$

Given the privacy parameter α, the values of $halt$, gen satisfying all four equations simultaneously satisfy α-length-indistinguishability. If we set $halt=gen$ and solve this system of equations, we observe that the following is a solution (which we call the symmetric solution in the main text):

$$halt = gen = \frac{1}{e^\alpha + 1}$$

Another solution to the same system of equations, which we call the asymmetric solution, is desirable when input sequences X, Y are longer and therefore a smaller $halt$ing probability is preferable:

$$0 < halt < \frac{1}{e^\alpha + 1} \quad \text{and} \quad 1 - e^\alpha \cdot halt \le gen \le 1 - \frac{halt}{e^\alpha}$$

$\square$

4.2 Content Indistinguishability

Next, we prove that for the above choice of parameters, Φ_{SEQ} satisfies α-content-indistinguishability. That is, for any pair of true sequences X, Y of same length, the following holds:

$$\frac{\Pr[\Phi_{SEQ}(X) = S]}{\Pr[\Phi_{SEQ}(Y) = S]} \le e^{\alpha \cdot d_{\text{seq}}(X, Y)}$$

Proof. We divide into 3 possible cases depending on how $|S|$ relates to $|X| = |Y|$.

Case 1: $|S| = |X| = |Y|$. In this case, Algorithm 2 must have behaved as follows. Upon reaching its main loop (lines 5-14), it must have run lines 6-9 for $i = 1$ to *max_len* and, in each iteration, the event with probability $(1 - halt)$ must have occurred such that $X[i]$ was perturbed and the perturbed item was appended to S. The perturbation is performed by the function call to Algorithm 1,

which we denote by Φ_{EM}. Then, in iteration $i = max\_len + 1$, the event with probability $(1 - gen)$ must have occurred so that a sequence S with length exactly equal to $|X| = |Y| = n$ was returned. The odds-ratio probabilities of the overall run can be computed as:

$$\frac{\Pr[\Phi_{SEQ}(X) = S]}{\Pr[\Phi_{SEQ}(Y) = S]} \overset{?}{\lessgtr} e^{\alpha \cdot d_{seq}(X,Y)}$$

$$\frac{(1 - gen) \cdot \prod_{i=1}^{n}(1 - halt) \cdot \Pr[\Phi_{EM}(X[i]) = S[i]]}{(1 - gen) \cdot \prod_{i=1}^{n}(1 - halt) \cdot \Pr[\Phi_{EM}(Y[i]) = S[i]]} \overset{?}{\lessgtr} e^{\alpha \cdot d_{seq}(X,Y)}$$

$$\frac{\prod_{i=1}^{n} \Pr[\Phi_{EM}(X[i]) = S[i]]}{\prod_{i=1}^{n} \Pr[\Phi_{EM}(Y[i]) = S[i]]} \overset{?}{\lessgtr} e^{\alpha \cdot d_{seq}(X,Y)}$$

By Theorem 1, for any index i:

$$\frac{\Pr[\Phi_{EM}(X[i]) = S[i]]}{\Pr[\Phi_{EM}(Y[i]) = S[i]]} \leq e^{\alpha \cdot d(X[i], Y[i])} \tag{14}$$

Applying Equation 14, we obtain:

$$\prod_{i=1}^{n} e^{\alpha \cdot d(X[i], Y[i])} \overset{?}{\lessgtr} e^{\alpha \cdot d_{seq}(X,Y)}$$

$$e^{\alpha \cdot \sum_{i=1}^{n} d(X[i], Y[i])} \leq e^{\alpha \cdot d_{seq}(X,Y)}$$

which holds by definition of d_{seq}, completing the proof for Case 1.

Case 2: $|S| < |X| = |Y|$. In this case, Algorithm 2 must have run lines 6-9 for $|S| + 1$ iterations; in the first $|S|$ iterations, the event with probability $(1 - halt)$ must have occurred to build the perturbed sequence S, and in the last iteration the *halt*ing event must have occurred so that the sequence of length $|S| = m$ was returned without adding more elements. The odds-ratio becomes:

$$\frac{\Pr[\Phi_{SEQ}(X) = S]}{\Pr[\Phi_{SEQ}(Y) = S]} \overset{?}{\lessgtr} e^{\alpha \cdot d_{seq}(X,Y)}$$

$$\frac{halt \cdot \prod_{i=1}^{m}(1 - halt) \cdot \Pr[\Phi_{EM}(X[i]) = S[i]]}{halt \cdot \prod_{i=1}^{m}(1 - halt) \cdot \Pr[\Phi_{EM}(Y[i]) = S[i]]} \overset{?}{\lessgtr} e^{\alpha \cdot d_{seq}(X,Y)}$$

$$\frac{\prod_{i=1}^{m} \Pr[\Phi_{EM}(X[i]) = S[i]]}{\prod_{i=1}^{m} \Pr[\Phi_{EM}(Y[i]) = S[i]]} \overset{?}{\lessgtr} e^{\alpha \cdot d_{seq}(X,Y)}$$

Notice that this case is different than Case 1 in the bounds of the product—the product goes from $i = 1$ to length of $|S|$, which is shorter than $|X|$. Applying Equation 14:

$$e^{\alpha \cdot \sum_{i=1}^{m} d(X[i], Y[i])} \overset{?}{\lessgtr} e^{\alpha \cdot d_{seq}(X,Y)}$$

A distance metric d satisfies the non-negativity property by definition. Therefore, for $m < n = |X|$, we have:

$$\sum_{i=1}^{m} d(X[i], Y[i]) \leq \sum_{i=1}^{n} d(X[i], Y[i]) = d_{seq}(X, Y)$$

completing the proof for Case 2.

Case 3: $|S| > |X| = |Y|$. In this case, Algorithm 2 must have run lines 6-9 for $|X|$ iterations, and in each iteration the event with probability $(1 - halt)$ must have occurred. Then, for $|S| - |X|$ iterations, lines 10-13 must have run, and the item generation event with probability *gen* must have occurred in these iterations. Finally, to return S, a final iteration must have occurred with the

stopping event having $(1 - gen)$ probability. Let $m = |S|$ and $n = |X|$. The odds-ratio is:

$$\frac{\Pr[\Phi_{SEQ}(X) = S]}{\Pr[\Phi_{SEQ}(Y) = S]} \overset{?}{\lessgtr} e^{\alpha \cdot d_{seq}(X,Y)}$$

$$\frac{(1 - gen) \cdot \prod_{i=1}^{m-n} gen \cdot Pr[\text{random} = S[i]]}{(1 - gen) \cdot \prod_{i=1}^{m-n} gen \cdot Pr[\text{random} = S[i]]} \cdot \prod_{i=1}^{n}(1 - halt) \cdot \Pr[\Phi_{EM}(X[i]) = S[i]]} \overset{?}{\lessgtr} e^{\alpha \cdot d_{seq}(X,Y)}$$

where $Pr[\text{random} = S[i]]$ denotes the probability that the random sampling on line 11 of Algorithm 2 returns item $S[i]$. Note that the random sampling does not depend on the properties of inputs X, Y, therefore we can safely cancel most terms. We end up with:

$$\frac{\prod_{i=1}^{n} \Pr[\Phi_{EM}(X[i]) = S[i]]}{\prod_{i=1}^{n} \Pr[\Phi_{EM}(Y[i]) = S[i]]} \overset{?}{\lessgtr} e^{\alpha \cdot d_{seq}(X,Y)}$$

Applying Equation 14 and continuing in the same fashion as Case 1, it is straightforward to complete the proof. □