Local Differential Privacy Protocol for Making Key–Value Data Robust Against Poisoning Attacks

Hikaru Horigome[1], Hiroaki Kikuchi[1(✉)], and Chia-Mu Yu[2]

[1] Graduate School of Advanced Mathematical Science, Meiji University, 4-21-1 Nakano, Tokyo 164-8525, Japan
{cs212030,kikn}@meiji.ac.jp
[2] Department of Information Management and Finance, National Yang Ming Chiao Tung University, Hsinchu, Taiwan

Abstract. Local differential privacy is a technique for concealing a user's information from collectors by randomizing the information within the user's own device before sending it to unreliable collectors. Ye et al. proposed PrivKV, a local differential privacy protocol for securely collecting key–value data, which comprises two-dimensional data with discrete and continuous values. However, such data is vulnerable to a "poisoning attack," whereby a fake user sends data to manipulate the key-value dataset. To address this issue, we propose an Expectation-Maximization (EM) based algorithm, in conjunction with a cryptographical protocol for ensuring secure random sampling. Our local differential privacy protocol, called emPrivKV, offers two main advantages. First, it is able to estimate statistical information more accurately from randomized data. Second, it is robust against manipulation attacks such as poisoning attacks, whereby malicious users manipulate a set of analysis results by sending altered information to the aggregator without being detected. In this paper, we report on the improvement in the accuracy of statistical value estimation and the strength of the robustness against poisoning attacks achieved by applying the proposed method to open datasets.

Keywords: local differential privacy · key–value data · expectation maximization

1 Introduction

Our personal data are being used by many services such as item recommendation for online shops, personalized medical assistance, and fake user detection. For example, in a smartphone survey, users indicate their favorite apps such as $\langle \texttt{YouTube}, 0.5 \rangle$, and $\langle \texttt{Instagram}, 0.2 \rangle$, by stating the total time they used each of the apps. These data were stored in a key–value database, whereby each "key" is an app title and its associated "value" is the rating of that app by a particular user. However, collecting this data poses a significant challenges.

V. Torra and Y. Narukawa (Eds.): MDAI 2023, LNAI 13890, pp. 241–252, 2023.
https://doi.org/10.1007/978-3-031-33498-6_17

Local Differential Privacy (LDP) is one approach to addressing the challenge. Here, each user locally perturbs their personal data before sending it to an (untrusted) server. Many LDP protocols have been proposed for different types of data, including Erlingsson et al. [7] proposed an LDP. Ye et al. [1] proposed PrivKV, an LDP scheme that securely collects key–value data, two-dimensional data with discrete and continuous values. Other LDP protocols [2] [3] for key–value data have also been proposed.

However, because the perturbation is being performed locally, LDP protocols are vulnerable to "poisoning attacks," whereby an attacker injects fake users who send fake data for a target key, aiming to manipulate the server's analytical results such as the frequency of particular keys or their mean reputation scores. If a fake user sends fake key and value data without following the predetermined LDP protocol, the server would not be able to detect these data because of the privacy guarantee of LDP. Cao et al. [4] studied poisoning attacks on LDP schemes. Wu et al. [5] identified three types of poisoning attacks for PrivKV and demonstrated that PrivKV is vulnerable to these types of attacks. They also proposed defense methods against poisoning attacks. However, these methods require long-term observation of the collection of the data.

In this paper, we address the issues of poisoning attacks on the LDP protocol for key–value data. First, we use a cryptographical protocol called oblivious transfer (OT) [6] to prevent fake users from choosing keys intentionally. Instead of performing random sampling locally, our protocol ensures that the server is involved jointly in the secure sampling process. Second, we claim that the estimation algorithm used in PrivKV is the source of its vulnerability to poisoning. Because it is computed using a single frequency for a key, it is easily to manipulated when the number of targeted keys is small. Instead, we address this limitation by using an Expectation Maximization (EM) algorithm [8]. Because EM estimates posterior probabilities iteratively, so that the estimated probabilities are more consistent across all observed values, it can improve the accuracy when the number of users is large and much observed data are available.

To investigate whether our proposed protocol is robust against various types of poisoning attacks, we conducted experiments using both synthetic data and open datasets. The results enable us to compare our proposed scheme with the conventional schemes such as PrivKV and PrivKVM.

Our contributions are as follows.

- We propose a new LDP algorithm that is robust against some types of poisoning attacks. Our proposed algorithm improves the accuracy of estimates based on the iterative process of Bayesian posterior probabilities and preserves the statistics against poisoning data.
- We show the experimental results that show the robustness of the proposed protocol using both synthetic data and open data. The results show that the proposed method performs better than the PrivKV protocol in estimation accuracy and in robustness against poisoning attacks.

2 Local Differential Privacy

2.1 Fundamental Definition

Suppose that users periodically submit their location data to a service provider. Differential privacy guarantees that the randomized data do not reveal any privacy disclosure from these data. By contrast, LDP needs no trusted party in providing the guarantee. LDP is defined as follows.

Definition 1. *A randomized algorithm Q satisfies ϵ-local differential privacy if for all pairs of values v and v' of domain V and for all subset S of range Z ($S \subset Z$), and for $\epsilon \geq 0$, $Pr[Q(v) \in S] \leq e^{\epsilon} Pr[Q(v') \in S]$.*

2.2 PrivKV

PrivKV takes input data in the key–value from, a two-dimensional data structure of discrete ("key") and continuous ("value") variables, and estimates each key's frequency and its mean values. PrivKV's approach idea combines two LDP protocols, randomized response (RR) [13] for randomizing keys and value perturbation protocol (VPP) [12] for perturbing values. The dimension is restricted to two, but the key–value is known as a primitive data structure commonly used for several applications.

Sampling. Let S_i be a set of key–value tuples $\langle k, v \rangle$ owned by the i-th user. In PrivKV, the set of tuples is encoded as a d-dimensional vector, where d is the cardinality of the domain of keys K and a missing key is represented as $\langle k, v \rangle = \langle 0, 0 \rangle$. For instance, a set of key–values $S_i = \{\langle k_1, v_1 \rangle, \langle k_4, v_4 \rangle, \langle k_5, v_5 \rangle\}$ is encoded as a $d = 5$ dimensional vector $\boldsymbol{S}_i = (\langle 1, v_1 \rangle, \langle 0, 0 \rangle, \langle 0, 0 \rangle, \langle 1, v_4 \rangle, \langle 1, v_5 \rangle)$ where keys k_1, k_4 and k_5 are specified implicitly with 1 at the corresponding location. PrivKV performs 1-out-of-d random sampling to choose one element $\langle k_a, v_a \rangle$ from the d-dimensional vector $\boldsymbol{S}_i$ of key–value data.

Perturbing. The process has two steps: perturbing values and perturbing keys. It uses the VPP used in Harmony [12] for the chosen tuple. A value v_a in the key–value pair is discretized as $v'_a = \begin{cases} 1 & \text{with probability } (1 + v_a)/2, \\ -1 & \text{with probability } (1 - v_a)/2. \end{cases}$ The discretized value v' of the tuple $\langle 1, v_a \rangle$ is perturbed to give ${v^+}_a = VPP(v_a, \epsilon_2)$, defined as $v_a^+ = \begin{cases} v'_a & \text{w/p. } p_2 = e^{\epsilon_2}/(1 + e^{\epsilon_2}), \\ -v'_a & \text{w/p.} q_2 = 1/(1 + e^{\epsilon_2}), \end{cases}$ where ϵ_2 is the privacy budget for values. The value of the "missing" tuple $\langle 0, 0 \rangle$ is replaced by ${v^+}_a = VPP(v'_a, \epsilon_2)$, where v'_a is chosen uniformly from $[-1, 1]$.

A key is perturbed by the RR scheme [13] as

$$\langle k_a^*, v_a^+ \rangle = \begin{cases} \langle 1, v_a^+ \rangle & w/p.\ p_1 = \frac{e^{\epsilon_1}}{1+e^{\epsilon_1}}, \\ \langle 0, 0 \rangle & w/p.\ q_1 = \frac{1}{1+e^{\epsilon_1}}, \end{cases}$$

where v_a^+ is perturbed as described above. A "missing" tuple $\langle 0, 0\rangle$ is randomized as

$$\langle k_a^*, v_a^+\rangle = \begin{cases} \langle 0, 0\rangle & w/p.\ p_1 = \frac{e^{\epsilon_1}}{1+e^{\epsilon_1}}, \\ \langle 1, v_a^+\rangle & w/p.\ q_1 = \frac{1}{1+e^{\epsilon_1}}. \end{cases}$$

Each user submits the perturbed tuple $\langle k_a^*, v_a^+\rangle$ together with the index a of the tuple.

Estimating. Let f_i be a true frequency of key k_i and let f_i' be the observed key frequencies among the perturbed vectors, for which $k_i = 1$. We can have the maximum likelihood estimation (MLE) of the frequency as $\hat{f}_i = \frac{n(p-1)+f_i'}{2p_1-1}$, where $p_1 = \frac{e^{\epsilon_1}}{1+e^{\epsilon_1}}$.

From the compositional theorem of differential privacy [9], the sequential composition of randomized algorithms with privacy budgets ϵ_1 (for keys) and ϵ_2 (for values) is $(\epsilon_1 + \epsilon_2, 0)$-differential private.

2.3 Poisoning Attack

We assume that an attacker is able to inject m fake users into a system. The attacker has access to open information about the target LDP scheme, such as its privacy budget ϵ and perturbation procedure. With n genuine users, the server estimates the frequencies and the mean values for r target keys among the $n + m$ users. The attacker aims to intentionally manipulate the estimated frequency and mean value for the set of targeted keys. We assume that the attacker targets r keys out of d, aiming maximize the manipulation in terms of frequencies and mean values.

Wu et al. [5] proposed the following three types of poisoning attacks;

1. Maximal Gain Attack (M2GA). All fake users craft the optimal fake output of perturbed message so that both the frequency and mean gains are maximized, i.e., they choose a target key k (a random key out of r targeted keys) and send $\langle 1, 1\rangle$ to the server.
2. Random Message Attack (RMA). Each fake user picks a message uniformly at random from the domain and sends $\langle 0, 0\rangle$, $\langle 1, -1\rangle$, $\langle 1, 1\rangle$, with probabilities 1/2, 1/4, and 1/4, respectively.
3. Random Key–Value Pair Attack (RKVA). Each fake user picks a random key k from a given set of target keys, with a designated value of 1, and perturbs $\langle 1, 1\rangle$ according to the protocol.

Wu et al. [5] proposed two methods to detect fake users, (1) one-class classifier-based detection, where observations of multiple rounds for each user gives the feature vector used for outlier detection, which can distinguish between genuine and fake groups. (2) anomaly score based detection, where the anomalous behavior of sending the same key in multiple rounds is detected from the frequencies of keys in multiple rounds for each user. They reported that these defense methods are effective when the number of targeted keys is small. However, their methods assume that each user sends data in multiple rounds, implying that realtime detection would not be feasible.

3 Proposed Algorithm

3.1 Idea

To prevent attacker from poisoning fake key–value data, we propose two defense methods, a perturbation with OT (see Sect. 3.2) and an EM estimation for frequency and mean values (see Sect. 3.3).

First, we note that a poisoning attempt to increase the frequencies of target keys is performed by the intentional choice of keys without random sampling. Therefore, if the server performs the random sampling on the behavior of fake users, the poisoning attempt would fail. Even if the server chooses a random key, no information of the key–value data is compromised. Note that privacy budgets (ϵ_1 and ϵ_2) are spent only for perturbing keys and values. In this way, we ensure a secure sampling using a cryptographical protocol (OT).

Second, we consider the reasons why the estimation might have been subject to a poisoning attack. We claim that the MLE used in PrivKV has low estimate accuracy for a biased distribution because it is computed on the single frequency for a key. It is therefore vulnerable when the number of targeted keys is small. Instead, we attempt to address this limitation by using the EM algorithm. Because EM estimates posterior probabilities iteratively, giving estimated probabilities that are more consistent with all observed values, it can improve the accuracy when the number of users n is large and much observed data are available.

Table 1 summarizes our approach for each of the steps in PrivKV, that involve sampling, perturbing, and estimating.

Table 1. Comparison of defenses approaches

step	PrivKV [1]	Our work
1 Pre-sampling	1-out-of-d sampling	–
2 Perturbing	Value $VPP(v, \epsilon_2)$ Key $RR(k, \epsilon_1)$	
3 Post-sampling	–	**1-out-of-d OT**
4 Estimating	MLE	**EM**

3.2 Oblivious Transfer

An OT is a two-party cryptographical protocol whereby a sender transfers one of many pieces of information to a receiver, but remains oblivious as to which of the pieces has been sent.

Naor and Pinkas [6] proposed an 1-out-of-N OT protocol using the 1-out-of-2 OT as a building blocks, as follows.

1-out-of-N OT [6] Suppose A has N messages $m_0, \ldots, m_N \in \{0,1\}^n$, where $N = 2^\ell - 1$.

1. A generates 2ℓ secret key pairs $(K_1^0, K_1^1), \ldots, (K_\ell^0, K_\ell^1)$.
2. A sends to B the ciphertexts $C_0, \ldots, C_N$, where $C_I = m_I \oplus F_{K_1^{I_1}}(I) \oplus \cdots \oplus F_{K_\ell^{I_\ell}}(I)$ and I is the ℓ-bit string $I_1 \ldots I_\ell \in \{0,1\}^2$ and F_K is a pseudo-random function.
3. A and B perform ℓ 1-out-of-2 OT (K_i^0, K_i^1) so that B learns $K_1^{t_1}, \ldots, K_\ell^{t_\ell}$ where t is the index that B chooses from N messages such that $t = 1_1 \ldots t_i \in \{0,1\}^\ell$.
4. B decrypts C_t using $K_1^{t_1}, \ldots, K_\ell^{t_\ell}$ to obtain m_t.

We aim to prevent an M2GA attack where fake users intentionally choose a target key (or set of keys) with aim of increasing the frequency and the mean value of the particular targeted keys. Simply, we replace the 1-out-of-d random sampling of PrivKV by an 1-out-of-d OT protocol performed between the user (A in OT) with d key–value pairs and the server (B), which chooses one element $\langle k_a, v_a \rangle$. However, the server cannot perform the subsequent perturbing steps because it must learn neither whether the user has key k_a nor the private value $v_a \in [0,1]$. Therefore, we change the order of steps so that users perturb the keys and values before the server chooses randomly a key–value pair via OT.

Algorithm 1 describes the proposed perturbation process using OT protocol for sampling. The perturbed key–value pairs will be used for estimating the frequency and the mean for the keys. With the reordering of steps, users have to perturb key–value pairs for all d keys, which will increase the computational cost on the user side by a factor of d. We regard this increase in computation cost as negligibly small because perturbation is a lightweight process in comparison with the cryptographical cost of the 1-out-of-d OT. The algorithm is robust against poisoning attacks.

Proposition 1. *An M2GA poisoning attack against the PrivKV scheme with 1-out-of-d OT for sampling key–value pairs has the frequency and the mean gains as large as an RMA poisoning attack has.*

Proof. Using an OT protocol, the fake users in the M2GA attack are not able to intentionally select the targeted keys. They may craft an arbitrary value but the server can detect invalid pairs other than the valid perturbed pairs $\langle 0, 0 \rangle, \langle 1, -1 \rangle$ and $\langle 1, 1 \rangle$. Therefore, they can prepare the valid perturbed pairs with arbitrary fractions, which is equivalent to an RMA attack. Therefore, the frequency and the mean gains will be less than or equal to those of an RMA attack.

3.3 EM Estimation for Key–Value Data

The EM algorithm performs an iterative process whereby posterior probabilities are updated through Bayes' theorem [8]. We propose using the EM algorithm

Algorithm 1. Perturbation of key–value pairs with OT

$S_1, \ldots, S_n \leftarrow$ key–value data for n users.
for all $u \in \{1, \ldots, n\}$ **do** perturbs all $\langle k_a, v_a \rangle \in S_u$
  $v_a^+ \leftarrow VPP(v'_a, \epsilon_2)$ and $k_a^* \leftarrow RR(k'_a, \epsilon_1)$
  u with $\langle v_1^+, k_1^* \rangle, \ldots, \langle v_d^+, k_d^* \rangle$ performs 1-out-of-d OT with a server.
end for return The server has n perturbed key–value pairs.

Algorithm 2. EM algorithm for PrivKV

$\langle v^+, k^* \rangle \ldots \leftarrow$ the perturbed key–value pair for n users.
$\Theta^{(0)} \leftarrow$ a uniform probability for $X = \{\langle 1, 1 \rangle, \langle 1, -1 \rangle, \langle 0, 1 \rangle, \langle 0, -1 \rangle\}$.
repeat(E-step)
  $t \leftarrow 1$
  Estimate posterior probability $\hat{\theta}_{u,i}^{(t)} \leftarrow Pr[x_i|z_u] = \frac{Pr[z_u|x_i]\theta_i{}^{(t-1)}}{\sum_{s=1}^{|X|} Pr[z_u|x_s]\theta_s{}^{(t-1)}}$,
  (M-step) Update marginal probability $\theta^{(t)} \leftarrow \frac{1}{n}\sum_{u=1}^{n} \hat{\theta}_u^{(t-1)}$.
until $|\theta_i^{(t+1)} - \theta_i^{(t)}| \leq \eta$
for all $a \in K$ **do** estimate
  $\hat{f}_a \leftarrow n(\theta_{\langle 1,1 \rangle}^{(t)} + \theta_{\langle 1,-1 \rangle}^{(t)})$ and $\hat{m}_a \leftarrow \frac{\theta_{\langle 1,1 \rangle}^{(t)} - \theta_{\langle 1,-1 \rangle}^{(t)}}{\theta_{\langle 1,1 \rangle}^{(t)} + \theta_{\langle 1,-1 \rangle}^{(t)}}$
end for return $\hat{f}_1, \hat{m}_1, \ldots, \hat{f}_d, \hat{m}_d$

for estimating the frequency and mean values from key–value data perturbed in PrivKV.

Algorithm 2 shows the overall process for the proposed EM algorithm for estimating the frequency and means of key–value data. Given n perturbed values $z_1, \ldots, z_n$, we iterate the estimation of posterior probabilities for $x_1, \ldots, x_d$ as $\Theta^{(t)} = (\theta_1{}^{(t)}, \theta_2{}^{(t)}, \ldots, \theta_d{}^{(t)})$ until convergence.

4 Evaluation

4.1 Data

Our synthetic data comprises a Key–value data for each of three distributions: Gaussian ($\mu = 0, \sigma = 10$), Power-law ($F(x) = (1 + 0.1x)^{-\frac{11}{10}}$), and Linear ($F(x) = x$). Table 2 gives the means and variances of the synthetic data, where $d = 50$ distinct keys are evaluated for $n = 10^5$ users. Table 3 shows the statistics for the two open datasets used in our experiments.

4.2 Methodology

Accuracy Metrics. Given a set of key–value data provided by n users, we use emPrivKV, PrivKV, and PrivKVM(c=3) to estimate the frequency of key k, $\hat{f}_k$, and the mean value for k, $\hat{m}_k$. The Mean Square Error (MSE) for these estimates are defined as $MSE_f = \frac{1}{|K|}\sum_{i=1}^{|K|}(\hat{f}_i - f_i)^2, MSE_m = \frac{1}{|K|}\sum_{i=1}^{|K|}(\hat{m}_i - m_i)^2$, where f_k and m_k are the real frequency and mean for key k. After repeating each estimation 10 times, we evaluate the estimation accuracy.

Table 2. Synthetic Data ($n = 10^5$, $d = 50$)

distribution	$E(f_k/n)$	$Var(f_k/n)$	$E(m_k)$	$Var(m_k)$
Gaussian	0.49506	0.10926	−0.00987	0.43702
Power-law	0.20660	0.06290	−0.58681	0.25160
Linear	0.51	0.08330	0	0.34694

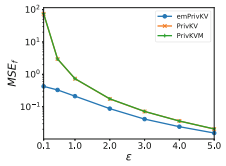
Table 3. Open datasets

item	MoveiLens [10]	Clothing [11]
# ratings	10,000,054	192,544
# users (n)	69,877	9,657
# items (d)	10,677	3,183
value range	0.5 – 5	1 – 10

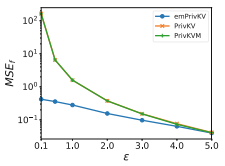
Robustness Metrics. The estimation algorithm is *robust* against poisoning attacks if a poisoning attack fails to alter the estimation results. We quantify the robustness via *frequency gain* as the sum of the distance between the estimated and the poisoned frequency for the key, i.e., the frequency gain is $G_f(Y) = \sum_{k \in T} E[\Delta \hat{f}_k]$, where $\Delta \hat{f}_k = \tilde{f}_k - \hat{f}_k$ is the distance and $\tilde{f}_k$ is the estimated frequency when key k is targeted by a poisoning attack. Similarly, the *mean gain* is the sum of the distance between the estimated and the poisoned value, defined as $G_m(Y) = \sum_{k \in T} E[\Delta \hat{m}_k]$ where $\Delta \hat{m}_k = \tilde{m}_k - \hat{m}_k$, and $\tilde{m}_k$ is the estimated mean value when key k is targeted by a poisoning attack.

4.3 Experimental Results

Accuracy with respect toϵ. Figs. 1a, 1b, 2a and 2b show the MSE distributions of frequencies and mean values for the open datasets, MovieLens and Clothing, respectively. Note that the MSE for emPrivKV are the minimum for both datasets and all ϵ. The accuracies with respect to the conventional PrivKV and PrivKVM are better by a factor of 100–1000 for small $\epsilon = 0.1$.

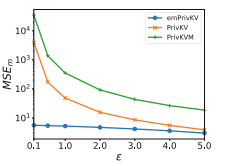


(a) MovieLens

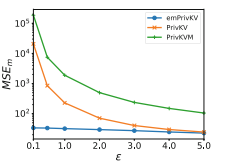


(b) Clothing

Fig. 1. MSE_f with respect to privacy budget ϵ



(a) MovieLens



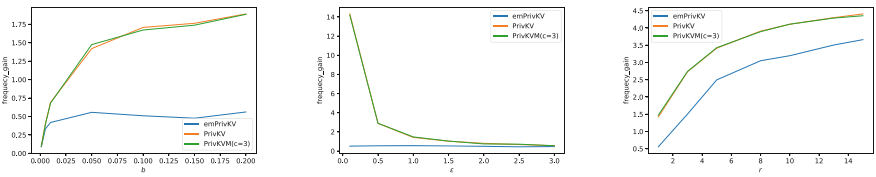
(b) Clothing

Fig. 2. MSE_m with respect to privacy budget ϵ

Frequency Gain. Figures 3a, 3b, 3c show the distributions of frequency gain with respect to the fraction of malicious users b, the privacy budget ϵ and the number of target key r, respectively, for the three types of poisoning attacks (M2GA, RMA and RKVA), when using the synthetic data (Gaussian distribution).

Note that an M2GA (see Figs. 3a, and 3b) causes the greatest gains for the three poisoning schemes. This is to be expected because it makes the strongest assumption (i.e., that malicious users are able to control the output arbitrarily) and therefore represents the greatest risk to LDP schemes.

The emPrivKV results show almost always the least gain for all types of poisoning attack and all parameters b, ϵ and r. As the fraction of malicious users b increases, the gains for PrivKV increase accordingly (see Fig. 3a). By contrast, the gain of emPrivKV is stable at 0.5. The gain of emPrivKV for $b = 0.2$ is 70.3% of PrivKV. Therefore, it is more robust against the worst type of poisoning attack (M2GA).



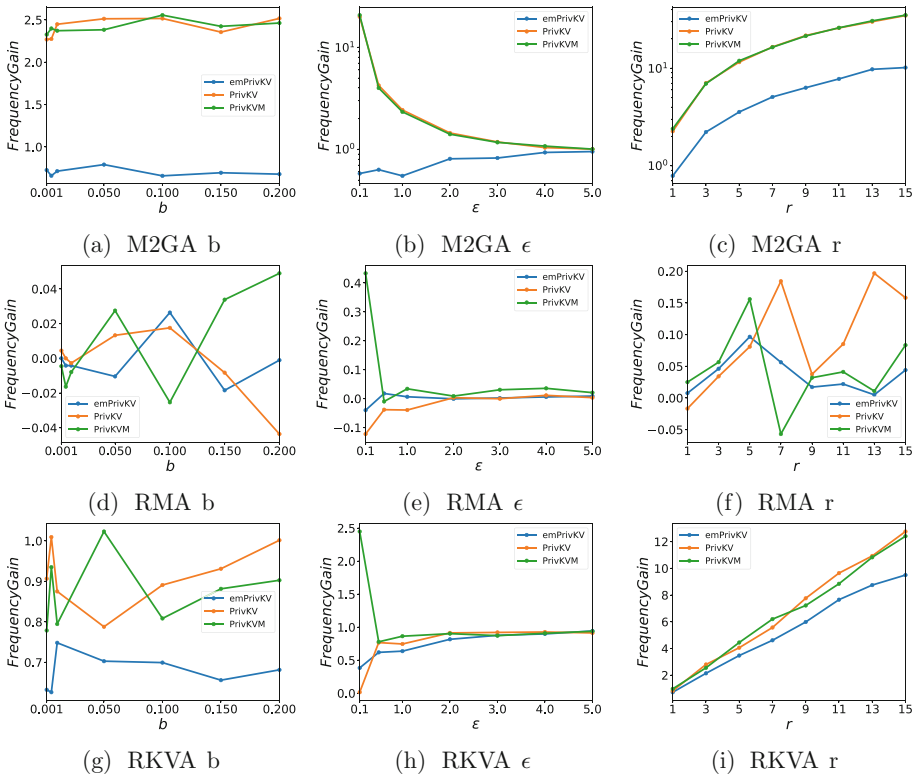
(a) M2GA b (b) M2GA ϵ (c) M2GA r

Fig. 3. Frequency gain for poisoning attacks(Gaussian)

Figure 4 shows the frequency gains for the MovieLens dataset. The gains distributions are similar to those using the Gaussian synthetic data, except for the effect the fraction-of-malicious-users parameter b (see Figs. 4a and 4g). The gain does not depend on b for M2GA (Fig. 4a), and is unstable for RKVA (Fig. 4g). The MovieLens data shows greater gains than the synthetic data (by a factor of 2–5) because the keys are not distributed as for the Gaussian distribution and there are many low-frequency keys (such as minor movie titles with very small audiences). These low-frequency keys are more vulnerable to low-resource poisoning attacks. With the same number of malicious users, the manipulated keys were already saturated in the MovieLens dataset. Therefore, the gains are greater in this case than for the synthetic-data case.

Mean Gain. The emPrivKV had always smaller gain than the PrivKV and PrivKVM had. For example, the gain for emPrivKV at $b = 0.2$ is stable around 1.0, which is 1/3 of that for PrivKV and 1/10 of that for PrivKVM. We observe similar results for the three LDP schemes with the MovieLens dataset (see Fig. 5a). Here, PrivKVM is seen as the most vulnerable against poisoning attacks.

The emPrivKV has the smallest gain with respect to privacy budget ϵ, as shown in Figs. 5b. The mean gains increase for PrivKV and PrivKVM as ϵ decreases. By contrast, the gain for the emPrivKV stays low, i.e., showing only minimal effects from poisoning attacks. This demonstrates the robustness of



(a) M2GA b (b) M2GA ϵ (c) M2GA r

(d) RMA b (e) RMA ϵ (f) RMA r

(g) RKVA b (h) RKVA ϵ (i) RKVA r

Fig. 4. Frequency gains for poisoning attacks (MovieLens)

emPrivKV. The gain increases linearly with number of targeted keys r. Figures 5c and 5i show the linear increase of the mean gains. Note that emPrivKV has the least coefficient for all the LDP schemes.

The LDP schemes did not show the significant differences with respect to RMA poisoning. Figure 5d shows that the differences in gain increase as the fraction of malicious users b increases.

4.4 Discussion

The experimental results demonstrate that the emPrivKV scheme is more robust than other LDP schemes. There are three possible reasons for this.

First, the PrivKV is based on the MLE, where the single-highest frequency is regarded as the expected value of the perturbation. Therefore, the scheme is likely to be affected by manipulating the highest frequency. By contrast, the EM algorithm iteratively adjusts the probabilities based on all the observed frequencies. Therefore, even if the highest frequency has been manipulated, the other elements help to mitigate against the manipulation of frequency.

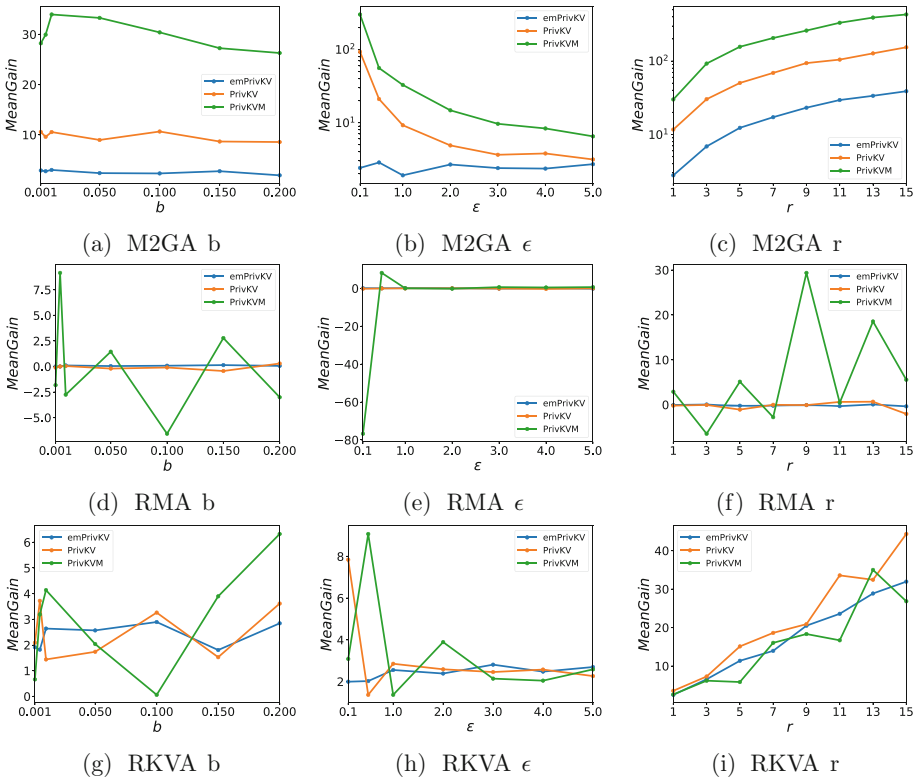


Fig. 5. Mean gain of poisoning attacks (MovieLens)

Second, we estimate the mean value based not only on the positive statistics ($v'_k = 1$) but also on both positive and negative statistics ($v'_k = 1$ and 0). This makes the estimation more robust against poisoning attacks and is the reason why the emPrivKV had a smaller mean gain.

Finally, based on our experimental results for gains, we can estimate the overall robustness of the proposed protocol. Following Proposition 1, M2GA is not relevant if perturbation with the OT protocol is used. Therefore, the gains from poisoning attacks on the proposed protocol can be estimated as the maximum of the gains for RMA and RKVA attacks (see Figs 4 and 5), as summarized in Table 4.

Table 4. Robustness against poisoning attacks (MovieLens, $b = 0.1$)

Attack	PrivKV [1]	Our work
Frequency gain	2.5 (M2GA)	0.7 (RKVA)
Mean gain	10 (M2GA)	3 (RKVA)

5 Conclusion

We have studied the privacy preservation of key–value data in the LDP algorithm PrivKV. Our proposed emPrivKV scheme uses the OT protocol for preventing intentional sampling of target keys and uses the EM algorithm for estimation. This makes the frequency and mean for keys robust against fake-data poisoning attacks. Our experiments using the MovieLens dataset, with the ratio of fake users to genuine users being 1 to 10, demonstrated that the proposed emPrivKV had a frequency gain of 0.7 and a mean gain of 3.0, which represent 28% (0.7/2.5) and 30% (3/10) of the gains for the PrivKV (fake users are 0.1 of genuine users), respectively. We conclude that the iterative approach works well for data perturbed via the LDP algorithm.

Acknowledgment. Part of this work was supported by JSPS KAKENHI Grant Number JP18H04099 and JST, CREST Grant Number JPMJCR21M1, Japan.

References

1. Ye, Q., Hu, H., Meng, X., Zheng, H.: PrivKV: key-value data collection with local differential privacy. IEEE S&P 294–308 (2019)
2. Gu, X., Li, M., Cheng, Y., Xiong, L., Cao, Y.: PCKV: locally differentially private correlated key-value data collection with optimized utility. In: USENIX Security Symposium, pp. 967–984 (2020)
3. Ye, Q., et al.: PrivKVM*: revisiting key-value statistics estimation with local differential privacy. IEEE Trans. Dependable Secure Comput. (2021)
4. Cao, X., Jia, J., Gong, N.Z.: Data poisoning attacks to local differential privacy protocols. In: USENIX Security Symposium, pp. 947–964 (2021)
5. Wu, Y., Cao, X., Jia, J., Gong, N.Z.: Poisoning attacks to local differential privacy protocols for key-value data. In: USENIX Security Symposium, pp. 519–536 (2022)
6. Naor, M., Pinkas, B.: Computationally secure oblivious transfer. J. Cryptol. **18**(1), 1–35 (2005)
7. Erlingsson, Ú., Pihur, V., Korolova, A.: RAPPOR: randomized aggregatable privacy-preserving ordinal response. In: ACM Conference on Computer and Communications Security, pp. 1054–1067 (2014)
8. Miyagawa, M.: EM algorithm and marginal applications. Adv. Stat. 16(1), 1–19. (in Japanese)
9. Dwork, C., Roth, A.: The algorithmic foundations of differential privacy. Found. Trends Theor. Comput. Sci. **9**(3–4), 211–407 (2014)
10. MovieLense 10M Dataset. https://grouplens.org/datasets/movielens/. Accessed 2022
11. Clothing Fit Dataset for Size Recommendation. https://www.kaggle.com/datasets/rmisra/clothing-fit-dataset-for-size-recommendation. Accessed 2022
12. Nguyên, T.T., Xiao, X., Yang, Y., Hui, S.C., Shin, H., Shin, J.: Collecting and analyzing data from smart device users with local differential privacy. arXiv:1606.05053 (2016)
13. Warner, S.L.: Randomized response: a survey technique for eliminating evasive answer bias. J. Am. Stat. Assoc. 63–69 (1965)