A Poisoning-Resilient LDP schema leveraging Oblivious Transfer with the Hadamard Transform

Masahiro Shimizu[1] and Hiroaki Kikuchi[1][0000−0002−0903−8430]

School of Interdisciplinary Mathematical Sciences, Meiji University.
4-21-1 Nakano, Tokyo 164-8525, Japan. kikn@meiji.ac.jp

Abstract. In recent years, Local Differential Privacy (LDP) has been actively used to collect and utilize users' usage history from smart devices with privacy considerations. However, since LDP allows users to add noise by themselves, Cao et al. pointed out that it is vulnerable to poisoning attacks where malicious users can intentionally manipulate data and send it to servers, thereby tamper with the aggregation results. Therefore, this study examines the application of an Oblivious Transfer (OT) protocol to the LDP protocol CMS to improve robustness against poisoning attacks. To address the challenge that the amount of data transmission and processing costs increase in proportion to the length of CMS's vector, we introduce the Hadamard Count Mean Sketch (HCMS) utilizing the Hadamard transform. The proposed method is experimentally implemented, and its security and efficiency are evaluated using open data.

Keywords: Local Differential Privacy, Count Mean Sketch, Hadamard Count Mean Sketch, Oblivious Transfer, Hadamard transform

1 Introduction

Increasingly, individuals are expressing concern about the extent to which their personal data is being utilized by digital platforms without their explicit consent. Users are becoming acutely aware of the vast amounts of personal information collected by online platforms, ranging from their browsing history and social media interactions to their location data and purchasing behavior. There is a growing sense of unease about how this data is being leveraged for targeted advertising, algorithmic profiling, and other commercial purposes, often without transparent disclosure or meaningful consent. The need to safeguard data privacy has emerged as a pressing concern for users seeking greater transparency, accountability, and control over their digital footprint.

Local differential privacy (LDP) technology serves as a mechanism to instill trust among users by implementing a randomized transformation of their personal data prior to its transmission to a server. This procedural step ensures that even untrusted servers are incapable of discerning the confidential values associated with individual users. The landscape of LDP encompasses a multitude of

proposed schemes, including the Randomized Response [12], the PrivKV [11], and the Google's RAPPOR [4].

Implementing local perturbation is a viable strategy to ensure data privacy; however, it remains susceptible to malicious actions carried out by clients. Cao et al. [1] demonstrated that the estimated statistics derived from an LDP scheme can be manipulated by a group of malicious clients. This presents a critical concern that necessitates attention in the realm of data privacy. Wu et al. [9] studied poisoning in key-value data. Huang et al. [13] proposed a anti-poisoning measure framework, called *LDPGuard*, observing statistical differences.

This paper delves into the vulnerability of LDP when subjected to poisoning attacks, offering a protocol rooted in Apple's Count Mean Sketch (CMS) [3] Leveraging Oblivious Transfer (OT) [5], our protocol aims to thwart malicious clients from circumventing the randomization process. When the OT is applied to randomized process, clients are compelled to engage in the randomization step securely, with facilitated by a semi-honest server.

Nevertheless, OT poses a significant computational and communicational burden, necessitating multiple public-key encryption/decryption processes per bit of CMS vector. In this paper, to mitigate these expenses, we advocate for the integration of the Hadamard transform into the CMS protocol, capitalizing on its advantageous mathematical properties. The Hadamard transform exhibits uniform distribution of values within each row, mutual orthogonality among rows, and the transpose of the matrix is nearly its inverse. Leveraging these characteristics, we propose a novel LDP scheme that employs the Hadamard transform, offering efficiencies in communication costs and robust resilience against a spectrum of poisoning attacks.

The contributions of our study are as follows:

1. We introduce a novel LDP scheme, termed OT-HCMS, which leverages OT to enhance resilience against poisoning attacks.
2. We propose the utilization of the Hadamard Transform to mitigate the computational and communication overhead in OT, with minimal impact on accuracy.
3. We conduct experiments using open-data to assess the accuracy, quantified by mean estimation error, and the security against poisoning attacks, measured by adversary frequency gains, of the OT-HCMS scheme. Our experimental findings demonstrate the efficiency of the proposed protocol in terms of both accuracy and security.

2 Local Differential Privacy

2.1 Fundamental Definition

Let D and Z be sets of input and output values. Suppose that users periodically submit their location data to a service provider. Differential privacy guarantees that the randomized data do not cause 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 D 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 Count Mean Sketch

Count Mean Sketch (CMS) [3] takes input of a private value $d \in D$ and returns a binary vector of length m, where typically $m < |D|$. Let $H = \{h_j | h_j : D \to [m], j \in [k]\}$ be a set of k hash functions. Given $d \in D$, each user uniformly chooses j-th hash functions and encodes d as m-dimensional vector $\boldsymbol{v} \in \{-1, 1\}^m$ where $h(d)$-th element is 1 and other $m-1$ elements are -1. Each bit of $\boldsymbol{v} = (v_1, \ldots, v_m)$ is flipped with predetermined probability as

$$\tilde{v_i} = \begin{cases} v_i & w./p.\ p = \frac{e^{\epsilon/2}}{1+e^{\varepsilon/2}}, \\ -v_i & w./p.\ q = \frac{1}{1+e^{\varepsilon/2}}. \end{cases}$$

CMS randomization algorithm is known as ϵ-local differentially private.

The server-side of CMS estimates frequencies for $d_i \in D$ out of n users in the following ways. Let S be a set of perturbed data $\{(\tilde{\boldsymbol{v}}^{(1)}, j^{(1)}), \ldots, (\tilde{\boldsymbol{v}}^{(n)}, j^{(n)})\}$. For each $i \in \{1, \ldots, n\}$, server computes $\tilde{\boldsymbol{x}}^{(i)} = k(\frac{c_\epsilon}{2}\tilde{\boldsymbol{v}}^{(i)} + \frac{1}{2}\mathbf{1})$, where $c_\epsilon = \frac{e^{\epsilon/2}+1}{e^{\epsilon/2}-1}$ and $\mathbf{1} = (1, \ldots, 1)$. Server computes a $k \times m$ *sketch* matrix M, where $M_{j^{(i)},\ell}$ is sum of $\tilde{x_1}^{(i)} + \cdots + \tilde{x_m}^{(i)}$ for all $i = 1, \ldots, n$. Finally, given sketch M, the server estimates the frequency for $d \in D$ as a mean of k counts,

$$\tilde{f}(d) = \left(\frac{m}{m-1}\right)\left(\frac{1}{k}\sum_{\ell=1}^{k} M_{\ell,h_\ell(d)} - \frac{n}{m}\right). \tag{1}$$

2.3 Hadamard Count Mean Sketch

Let K be a power of two and $H_K \in \{\pm 1\}^{K \times K}$ be the Hadamard matrix of size $K \times K$. Let $H_1 = (1)$ and for $m = 2^i$

$$H_m = \begin{pmatrix} H_{m/2} & H_{m/2} \\ H_{m/2} & -H_{m/2} \end{pmatrix}.$$

The Hadamard matrices satisfies the following properties (i) The number of $+1$'s in each row (except the first) is $K/2$, (ii) Any two rows agree on exactly $K/2$ locations, (iii) Rows are mutually orthogonal, and (iv) The transpose is closely its inverse, i.e., $H_K H_K^T = K I_K$.

The Hadamard basis transform can be used to spread information from a sparse vector where only a single element is 1. The transform is performed as $\boldsymbol{w} = H_m \boldsymbol{v}$, where H_m is the Hadamard matrics. Hadamard Count Mean Sketch (HCMS) is a randomization mechanism where a client send a single bit to server without sending $m-1$ redundant elements of -1, as following steps.

This algorithm also satisfies ϵ-local differentially private.

Algorithm 1 Hadamard Count Mean Sketch (HCMS) [3]

Require: $d \in D$, n clients, a server, parameters ϵ, k, m.

1. Client-side
   (a) For $i \in \{1, \ldots, n\}$, i-th client chooses uniformly j-th hash function h_j. and sets an m-dimensional vector $\boldsymbol{v} = (v_1, \ldots, v_m)$ where $v_{h_j(d)} = 1$ and $v_j = 0$ for $j \neq h_j(d) \in [m]$.
   (b) The client transform $\boldsymbol{w} = H_m \boldsymbol{v}$ and uniformly samples $\ell^{(i)} \in [m]$.
   (c) The client flips $w_{\ell^{(i)}}$ with probability $p = \frac{e^\epsilon}{e^\epsilon + 1}$ as

$$\tilde{w}^{(i)} = \begin{cases} w_{\ell^{(i)}} & w./p.\ p, \\ -w_{\ell^{(i)}} & w./p.\ 1-p. \end{cases}$$

   and sends tuple $(\tilde{w}^{(i)}, j^{(i)}, \ell^{(i)})$ to server.

2. Server-side
   (a) Given perturbed data $(\tilde{w}^{(1)}, j^{(1)}, \ell^{(1)}), \ldots, (\tilde{w}^{(n)}, j^{(n)}, \ell^{(n)})$, a server computes sketch matrix $M \in R^{k \times m}$ such that for $j = 1, \ldots, k$ and $\ell = 1, \ldots, m$,

$$M_{j,\ell} = \sum_{j=j^{(i)}, \ell=\ell^{(i)}} k c_\epsilon \tilde{w}^{(i)}$$

   where $c_\epsilon = \frac{e^\epsilon + 1}{e^\epsilon - 1}$.
   (b) The server transforms as $M' = M H_m^T$ for which the frequency for $d \in D$ is estimated in Eq. (1).

2.4 1-out-of-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. Algorithm 2 shows a known construction [5] using RSA encryption.

2.5 Poisoning

Wu et al. [9] proposed the following three types of poisoning attacks;

1. Maximal Gain Attack (MGA). 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 item (a random key out of m targeted item) and send the fake data to the server.
2. Random Message Attack (RMA). Each fake user picks a message uniformly at random from the domain and sends it with according probabilities.
3. Random Input (Key-Value Pair) Attack (RIA). Each fake user picks a random item from a given set of target item, with a designated value of 1, and perturbs it according to the protocol.

Algorithm 2 1-out-of-2 Oblivious Transfer[5]

Require: message m_0, m_1
Sender generates RSA key pair private key d, public keys N, e
Sender sends public keys to Receiver
Sender has two random message x_0, x_1

1. Sender sends x_0, x_1 to Receiver
2. Receiver chooses $b \in \{0, 1\}$ and generates random k and computes $v = (x_b + k^e) \mod N$ the encryption of k, blind with x_b. Receiver sends v to Sender.
3. Sender computes $k_0 = (v - x_0)^d \mod N$, $k_1 = (v - x_1)^d$ and $m'_0 = (m_0 + k_0) \mod N, m'_1 = (m_1 + k_1) \mod N$ Sender send m'_0, m'_1.
4. Receiver computes $m_b = (m'_b - k) \mod N$.

Ensure: m_b

Wu et al. [9] 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 Method

3.1 Threat model

LDP schemes are vulnerable against poisoning attack that aims to manipulate the estimated statistics. In CMS and HCMS, some probabilistic processes can be replaced by arbitrary intentional ones.

We assume that a fraction of clients, specified as β, are malicious and controlled arbitrarily as an adversary. According to the works [1], we define three typical poisoning attacks performed by a set of malicious clients.

In LDP, a server is semi-trusted in the sense it follows a predetermined protocol but is curious about the private value sent from client. Hence, the perturbation process is performed at the client side so that the server has no chance to learn the original value.

Random Perturb Attack RPA aims to disrupt the estimation by intentionally sending many random data so that the frequencies of items are failed to be computed very accurately. With this attack, malicious clients send fake tuple $(\tilde{w}^{(i)}, j^{(i)}, \ell^{(i)})$, where $\tilde{w}^{(i)} \in \{-1, 1\}, j^{(i)} \in \{1, \ldots, k\}$ and $\ell^{(i)} \in \{1, \ldots, m\}$ are chosen uniformly. All frequencies would be close to the mean.

Random Item Attack RIA aims to have a set of target items $T \subset D$ had lower frequencies than true one. In the RIA, malicious clients choose target item $t \in T$ and follow the LDP perturbation procedures, Steps 1a–1c.

Maximal Gain Attack MGA replaces the output of LDP perturbation process by the fake data without performing the legitimate perturbation. It aims to maximize the frequencies of the targeted item and hence is known as highest risk in LDP scheme.

Suppose that an adversary aims to increase the frequency of target item d. In CMS and HCMS, there are some steps determined probabilistic ways where malicious clients are allowed to manipulate without being detected. Potentially, output values

1. $j^{(i)}$ index of hash function,
2. $\ell^{(i)}$ index of element of m-dimensional vector $\boldsymbol{w}$,
3. $\tilde{w}^{(i)}$ perturbed value either 1 or -1

are vulnerable to be altered. In this work, we focus on $\tilde{w}$ because fake j and ℓ are easily detected and has smaller impact to the estimation than the manipulation of $\tilde{w}$. Hence, we assume that malicious clients follow Step 1a and 1b in Algorithm 1 as specified, but violate Step 1c as they like. To maximize the privacy gain defined of the difference of estimated frequencies with and without poisoning as $FG = \sum_{t \in T} \boldsymbol{E}[\tilde{f}_t - \hat{f}_t]$, malicious clients return fake tuple $(\tilde{w}^{(i)}, j^{(i)}, \ell^{(i)}) = (1, j, \ell)$ for uniformly chosen j and ℓ.

3.2 Secure OT-HCMS

To prevent malicious clients from poisoning attacks, we explore a simple countermeasure using an oblivious transfer. Malicious client craft $\tilde{w}^{(i)}$ that is inconsistent with Step 1c and $h_j(d)$ or ϵ. Therefore, we force them to follow the protocol based on 1-out-of-$(1/p)$ OT protocol between client and server.

Let $p = \frac{e^\epsilon}{e^\epsilon + 1}$ be $1/4$ and $\tilde{w}^{(i)} = 1$ for i-th client. The client (sender) has messages m_{00}, m_{01}, m_{10} and m_{11} as $1, 1, 1$ and -1, respectively. Then, the client and a server jointly peform 1-out-of-4 OT, with randomly shuffled messages. The server (receiver) picks $b \in \{00, 01, 10, 11\}$ and receives one of the messages that w is flipped with probability $1/4$ as specified.

With the OT and the security of public-key crypto-system, no malicious client has a chance to replace the output of perturbation and the semi-trusted server learns about the client's private value no more than a guarantee of ϵ-LDP. Therefore, attacks RPA and MGA are infeasible in the OT based CMS scheme. Adversary can perform only RIA, which has less significant impact to the the LDP estimation than MGA does.

However, the OT protocol is expensive in both communication and computation costs. Algorithm 2 requires one public-key encryption and two decryptions processes and the number of ciphertexts is proportional to the number of candidate messages, which is $\log(1/p)$. Additionally, the CMS client sends vector

$\tilde{\boldsymbol{v}} \in \{-1, 1\}^m$ that requires running OT protocol m times, which depends on the size of domain D. Therefore, OT-CMS is not very good idea.

Instead of the CMS with OT, we propose an OT-HCMS scheme where a client sends one bit $\tilde{w} \in \{-1, 1\}$ perturbed according to the privacy budget. Hence, we need to preserve the security of one bit. It reduces the number of OT protocols by one and saves considerable computation and communication costs. [3] proves that the expected value of frequency for item d is equal to that of the true estimation, but it may suffer estimation error according to privacy budget. In Section 4, we quantify the estimation error using some open data.

Algorithm 3 shows the procedure of our proposed secure LDP scheme.

Algorithm 3 Secure OT-HCMS

Require: $d \in D$, n clients, a server, parameters ϵ, k, m.
Require: $2^\tau = \lceil 1/p \rceil$ for $p = \frac{e^\epsilon}{e^\epsilon + 1}$.

1. same as Step (1a) in HCMS (Algorithm 1).
2. same as Step (1b) in HCMS.
3. i-th client prepares 2^τ messages of $\{-1, 1\}$ according to ϵ and performs 1-out-of-2^τ OT jointly with a server. The client sends $j^{(i)}$ and $\ell^{(i)}$ to the server.
4. The server receives $\tilde{w}^{(i)}$ through OT for $i = 1, \ldots, n$ and performs Step (2a) in HCMS.
5. same as Step (2b) in HCMS.

4 Evaluation

4.1 Datasets

In order to evaluate the security against poisoning and the accuracy of estimation of the proposed protocol, we conduct an experiment using open-data Click Stream[8]. It contains information on clickstream from online store offering clothing for pregnant women. It contains 165,474 records for 14 attributes. We use the attribute "clothing model" that has $|D| = 43$ distinct values. The most frequent item (A2) is purchased 3,013 times. Fig. 1 shows the distribution of frequencies of items in the dataset.

4.2 Methodology

Fig. 2 shows the list of default parameter used in our experiment. We apply LDP schemes, CMS, HCMS, OT-CMS and OT-HCMS to the dataset and observe the estimation for 50 times. In poisoning attacks, RMA, RIA and MGA, we repeat 10 attacks targeted to the set of items `A18` and `A34`, which are chosen as representative items with high frequency.

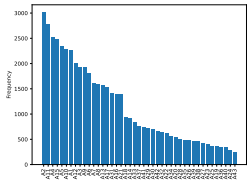

Fig. 1. Distribution of frequencies f_d for item d

parameter	value
privacy budget ϵ	1.0
vector size m	2^7
the number of hash functions k	2^{10}
the fraction of malicious clients β	0.01
the number of targeted items r	1

Fig. 2. Default Parameters

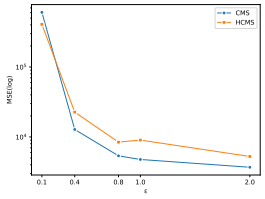

Fig. 3. MSE with respect to ϵ

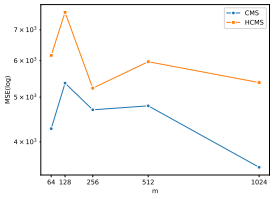

Fig. 4. MSE with respect to m

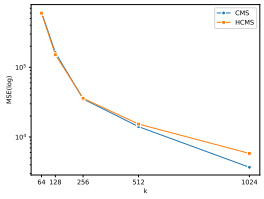

Fig. 5. MSE with respect to k

We evaluate the frequency estimation accuracy quantified in the Mean Squared Error (MSE) and the security against poisoning attacks via the Frequency Gain (FG) defined as $FG = \sum_{t \in T} \tilde{f}(t) - \hat{f}(t)$, where $\tilde{f}(t)$ and $\hat{f}(t)$ are the estimated frequency of item t without and with poisoning, respectively. Note that the gain is of adversary's viewpoint and the smaller gain implies more robust against poisoning.

4.3 Results

(1) Utility Figs. 3, 4, 5, and Table 1 show the MSE of CMS and HCMS with respect to privacy budget ϵ, vector size m (CMS) and the number of hash functions k.

We found that the proposed HCMS estimation has slightly greater MSE than that of CMS for all cases. The error by HCMS is 47.3% for $\epsilon = 1.0$, and is 21.7% in average, as shown in Fig.3. With varying vector size m, the MSEs are unstable due to the uneven distribution of payment records, shown in Fig. 4. The difference in error between with and without Hadamard transform are relatively small with respects to the number of hash function k (Fig. 5). Overall, we estimate frequencies in the HCMS with reduced communication overhead but with the reduced accuracy.

(2) Frequency Gains Figs. 6, 7, and 8, Tables 2, 3, and 4 show FGs with respect to privacy budgets ϵ, malicious client rates β, and the numbers of targeted

Table 1. MSEs in frequency estimation ($\times 10^3$)

ϵ	CMS	HCMS
0.1	604.66	407.07
0.4	12.78	22.63
0.8	5.37	8.43
1.0	4.76	9.03
2.0	3.69	5.26

m	CMS	HCMS
64	4.27	6.16
128	5.36	7.64
256	4.70	5.23
512	4.79	5.97
1024	3.52	5.38

k	CMS	HCMS
64	613.03	596.29
128	162.13	151.32
256	35.22	35.89
512	14.07	15.34
1024	3.68	5.81

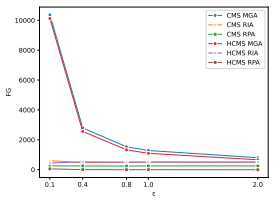

Fig. 6. FG with respect to ϵ

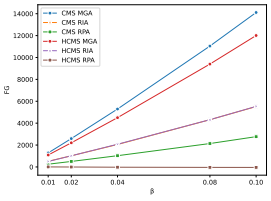

Fig. 7. FG with respect to malicious rate β

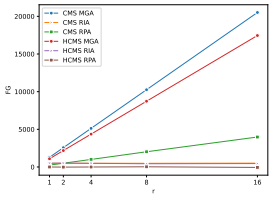

Fig. 8. FG with respect to # target items r

items r, respectively. We quantify the resiliences of CMS and HCMS against three kinds of poisoning, including MGA, RIA and RPA. Note that smaller FG means better robustness against attacks.

First, we note that MGA gives the greatest gain (FG) for three attacks, for any parameters ϵ, β, and r. From Table 2, RPA of CMS is 2.5 at $\epsilon = 1.0$, which is almost half of RPA ($FG = 5.08$), which is almost half of MGA ($FG = 12.82$). Similar effect can be seen for HCMS. It makes sense because of the definition of poisonings. Second, we found that FGs of HCMS are always smaller (16.0% in average) than that of CMS. This holds for all conditions of ϵ, β, and r. The difference between CMS and HCMS becomes maximum when MGA was performed. The HCMS reduces the gain of adversary by 21 ($\beta = 0.1$ in Table 3), which is about 15% of CMS. The possible reason why the robustness is improved in HCMS is that the Hadamard transform works the distribution of affected elements widely in all elements. Third, the differences in FG between CMS and HCMS varies with attacks.

(3) Security Improvement by OT We focus on the robustness enhancement given by the OT. Figs. 9 and 10 shows the distributions of mean FGs of CMS and OT-CMS, HCMS and OT-HCMS with respect to the fraction of malicious clients β. Obliviously, the OT protocol helps reducing adversary's gains significantly. Table 5 shows the mean FGs, where the improvement of security (FGs) ranges 51% – 75% for CMS and OT-CMS, and 47% – 58% for HCMS and OT-HCMS. With OT, malicious client is not able to skip perturbation process and the MGA is eventually reduced to the RIA attack.

Table 2. FG($\times 10^2$) with respect to privacy budget ϵ

ϵ	RPA		RIA		MGA	
	CMS	HCMS	CMS	HCMS	CMS	HCMS
0.1	2.51	0.53	6.01	4.46	103.70	101.24
0.4	2.47	0	4.27	5.17	27.87	25.58
0.8	2.40	−0.08	4.99	4.98	15.31	13.28
1.0	2.50	−0.05	5.08	5.00	12.82	10.91
2.0	2.50	−0.08	5.03	5.04	7.96	6.60

Table 3. FG($\times 10^2$) with respect to the fraction of malicious client β

β	RPA		RIA		MGA	
	CMS	HCMS	CMS	HCMS	CMS	HCMS
0.01	2.51	0.04	5.18	5.01	12.82	10.91
0.02	5.03	−0.05	10.19	10.21	25.92	22.06
0.04	10.28	−0.24	20.52	20.73	52.91	45.03
0.08	21.39	−0.40	43.15	43.18	110.44	93.99
0.10	27.67	−0.43	55.19	55.30	141.11	120.09

4.4 Discussion

According to the experimental results, we confirm that the effect of Hadamard transform reduces the communication overhead required for mitigation of poisoning. In Table 6, we summarize the accuracy, the security and the communication cost of the proposed protocol (OT-HCMS) in comparison with the conventional one (CMS[3]). We show the representative parameter for poisoning (MGA, $\epsilon = 1.0$, $\beta = 0.1$, $m = 2^6$).

We observe that the proposed OT-HCMS is secure against poisoning and reduces the adversarial gain (FG) about 10-times to that of plain CMS and HCMS. The cost of security enhancement is the increase of error (MSE), which is almost double of the CMS without Hadamard transform. However, the accuracy can be improved with the increase of number of client and it is not critical in practice.

Table 4. FG($\times 10^2$) with respect to the number of targeted items r

r	RPA		RIA		MGA	
	CMS	HCMS	CMS	HCMS	CMS	HCMS
1	2.50	−0.01	4.93	5.04	12.82	10.91
2	4.99	−0.17	5.09	5.07	25.64	21.82
4	10.02	−0.02	4.95	4.84	51.28	43.64
8	20.20	0.25	4.89	4.29	102.55	87.27
16	39.77	−0.58	5.10	4.56	205.11	174.54

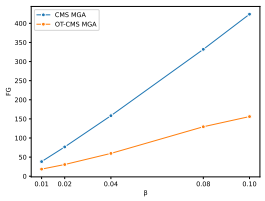

Fig. 9. FG of OT-CMS with respect to the fraction of malicious clients β

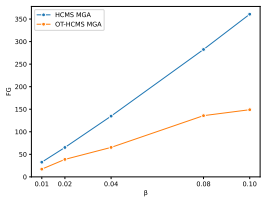

Fig. 10. FG of OT-HCMS with respect to the fraction of malicious clients β

Table 5. FGs of OT-CMS and OT-HCMS with respect to β

β	CMS	OT-CMS	HCMS	OT-HCMS
0.01	38.30	18.56	32.60	17.15
0.02	76.61	30.59	65.19	38.75
0.04	158.33	59.70	134.73	65.34
0.08	331.97	129.44	282.51	135.75
0.10	423.90	156.07	360.74	148.91

We also note that the Hadamard transform does not only contribute to save the communication, but also helps the reduction of FG from CMS (see FG 361 (OT-CMS) and 149 (OT-HCMS) in the Table 6). This is because that it works sampling uniformly over the transformed domain and estimates from mixed data sent from all (benign and malicious) clients. This could help somehow reducing robustness against poisoning.

5 Conclusions

We have studied a security enhancement of LDP protocol against the variety of poisoning attacks. The proposed scheme forces clients to join an oblivious transfer to perform value perturbation process according to the predetermined probability so that no malicious client skip the randomization step. We applied the Hadamard transform to the baseline protocol, the Count Mean Sketch, to

Table 6. Comparison

	CMS[3]	HCMS[3]	OT-CMS	OT-HCMS
Accuracy (MSE)	4.76	9.03	4.76	9.03
Security (FG)	1282	1091	361	149
Communication [s]	N/A	N/A	4.6	0.07

save the computational cost required for public-key encryption and the communication cost linear to the vector size by one. Our experiment using open-data demonstrate that the proposed protocol reduces the adversary's gain (FG) 47% – 58%, that is, it is robust against poisoning.

Our future studies include a security improvement for variety of DP schemes, an advanced scheme to mitigate other malicious behaviors performed at both client and server, and a new scheme to detect poisoning in shuffle model.

Acknowledgment. Part of this work was supported by JSPS KAKENHI Grant Number 23K11110 and JST, CREST Grant Number JPMJCR21M1, Japan.

References

1. X. Cao, J. Jia, N. Z. Gong, "Data poisoning attacks to local differential privacy protocols," USENIX Security Symposium, pp. 947-964, 2021.
2. Hikaru Horigome, Hiroaki Kikuchi and Chia-Mu, Yu, "Local Differential Privacy Protocol for Key-Value Data Robust against Poisoning Attacks," Modeling Decisions for Artificial Intelligence, pp.241-252, Volume 13890, 2023.
3. Differential Privacy Team, Leaning with Privacy at Scale(https://machinelearning.apple.com/research/learning-with-privacy-at-scale).
4. Úlfar Erlingsson, Vasyl Pihur, Aleksandra Korolova, "RAPPOR: Randomized Aggregatable Privacy-Preserving Ordinal Response", ACM Conference on Computer and Communications Security, pp.1054-1067, 2014.
5. Even, Shimo Goldreich, Oded Lempel, and Abraham, "A Randomized Protocol for Signing Contracts," Communications of the ACM, pp.205-210, 1982.
6. Gadotti, Andrea Houssiau, Florimond Annamalai, Meenatchi Montjoye, and Yves-Alexandre, "Pool Inference Attacks on Local Differential Privacy: Quantifying the Privacy Guarantees of Apple's Count Mean Sketch in Practice", 31st USENIX Security Symposium (USENIX Security 22), 2022.
7. J. C. Duchi, M. I. Jordan and M. J. Wainwright, "Local Privacy and Statistical Minimax Rates," 2013 IEEE 54th Annual Symposium on Foundations of Computer Science, Berkeley, CA, USA, pp. 429-438, 2013
8. clickstream data for online shopping, UCI Machine Learning Repository, 2019.
9. Y. Wu, X. Cao, J. Jia, and N. Z. Gong, "Poisoning Attacks to Local Differential Privacy Protocols for Key-Value Data, " USENIX Security Symposium, pp. 519-536, 2022.
10. M. Naor and B. Pinkas, "Computationally Secure Oblivious Transfer," Journal of Cryptology, Springer-Verlag, Vol. 18, No. 1, pp. 1-35, 2005.
11. Q. Ye, H. Hu, X. Meng, H. Zheng, "PrivKV : Key-Value Data Collection with Local Differential Privacy", IEEE S&P, pp. 294-308, 2019.
12. S. L. Warner, "Randomized response: A survey technique for eliminating evasive answer bias", Journal of the American Statistical Association, pp. 63-69, 1965.
13. K. Huang, et al., "LDPGuard: Defenses against Data Poisoning Attacks to Local Differential Privacy Protocols", IEEE Trans. on Knowledge and Data Engineering, pp. 1-14, 2024.