Computer Security Symposium 2022
24 - 27 October 2022

小さいテーブルを用いる 差分プライバシーのためのセキュアサンプリング

紀伊 真昇[1,a)]

概要：秘密計算でノイズをサンプリング（生成）することをセキュアサンプリングという．セキュアサンプリングを用いれば，ノイズの分布以外の情報を誰も得られないようにすることができる．セキュアサンプリングの手法として，事前に用意した暗号化済みテーブルを一様乱数で引く手法が以前から考えられている．この手法は安全性が高く計算量も少ないが，利用するテーブルが大きく通信量・メモリ使用量で他の手法に大きく劣る．本研究では暗号化したテーブルから $n(>1)$ 回引き出し，その和をノイズとして利用するプロトコルを提案する．同時に，このプロトコルで得られるノイズが差分プライバシーの意味での保護に利用できるようにするため，テーブルの生成アルゴリズムを提案する．$n=1$ の場合には 273GiB にまでなるテーブルを，$n=2$ の場合にこのアルゴリズムを用いれば 2.25MiB にまで縮小することができる．

キーワード：秘密計算，プライバシー保護，差分プライバシー，準同型暗号

A Secure Sampling Method with a Smaller Table for Differential Privacy

Kii Masanobu[1,a)]

Abstract: Secure sampling is method of sampling (generating) noise using a secret computation. Secure sampling ensures that no one can obtain any information other than the distribution of the noise. One of the secure sampling methods is to draw a pre-prepared encrypted table by uniform random numbers. Although this method is highly secure and at a low computational cost, it needs the large size of the tables used. In this study, we propose a protocol that draws $n(>1)$ times from the encrypted table and uses the sum as noise. At the same time, we propose an algorithm for generating tables in order to ensure that the noise obtained by the protocol can be used for differential privacy. The table size of 273GiB for $n=1$ can be reduced to 2.25MiB by using this algorithm for $n=2$.

Keywords: secure computation, privacy preservation, differential privacy, homomorphic encryption

1. 研究背景

秘密計算をはじめとするプライバシー保護のためのプロトコルでは，しばしばデータにノイズを加えることで情報を保護することがある。このときに利用されるノイズはほとんど完全に秘密でなくては，情報を保護することができない。利用するノイズについて唯一知られてよいのは，プロトコル参加者の間で予め共有される，ノイズが従う確率分布だけである。ノイズの正確な値はもちろん，そのノイズがある値以下である，といった部分的な情報も知られてはならない。この目的で用いられるプロトコルはセキュアサンプリングプロトコルと呼ばれる．

セキュアサンプリングの利用目的には，情報理論的安全性の達成と差分プライバシーの達成がある．前者の場合には一様分布に従うノイズは Vernam 暗号の鍵のように利用される．後者の差分プライバシーとは以下で定義される安全性であり，情報理論的安全性を緩和したものと見ること

[1] 日本電信電話株式会社 社会情報研究所
NTT Social Information Laboratories
[a)] masanobu.kii.gw@hco.ntt.co.jp

ができる [1].

定義 1.1 (Differential Privacy (DP))
2 つの実数 $\varepsilon(>0), \delta(\in[0,1])$ を考える．データベースの集合 $\mathcal{D}$ の 2 つの元 D_0, D_1 が「1 要素だけ異なる[†1]」とき，D_0 と D_1 は隣接しているという．

ランダム写像 $\mathcal{M}\colon \mathcal{D} \to X$ が，任意の隣接するデータベースの $D_0, D_1 \in \mathcal{D}$ と任意の部分集合 $S \subseteq X$ について次を満たすとき，$\mathcal{M}$ は (ε, δ) 差分プライバシーを達成している（あるいは与える）という．

$$\Pr[\mathcal{M}(D_0) \in S] \leq e^{\varepsilon} \Pr[\mathcal{M}(D_1) \in S] + \delta$$

差分プライバシーを達成しているランダム写像を差分プライベートメカニズム，あるいは単にメカニズムと呼ぶ． □

差分プライベートメカニズムとしては，決定的 (deterministic) な実数関数の出力にラプラス分布や正規分布に従うノイズを加えるものが代表的である．整数値を返す決定的関数に対しては，離散ラプラス分布 (discrete Laplacian distribution, [2]) [†2] に従うノイズを加えるメカニズムがよく用いられる．差分プライベートメカニズムに用いられるこれらのノイズは，平文では一様乱数から対数関数やビット演算等を用いて生成されるもので，秘密計算下でサンプルするのは容易ではない．

セキュアサンプリングの手法は大きく分けて次の 3 種類がある．

(1) 上記のような平文で行うサンプリング方法を秘密計算で実装する手法,

(2) 秘密計算の各々の参加者が平文でノイズをサンプリングして，秘密計算でそれらの和を計算する手法,

(3) 暗号化されたテーブル（配列）を事前に準備して秘密計算下でシャッフルした後，一様整数乱数でこのテーブルから引き出す手法.

(1) は安全に最適な誤差量[†3]のノイズをサンプルできるが，計算量がかなり大きい．(2) は計算・実装は簡単だが，各参加者がノイズの情報の一部を知ることが出来てしまうため，安全性を保つには誤差量が大きめのノイズを加える必要がある．(3) は安全で計算・実装は簡単だが，事前に準備するテーブルが公知であることなどから，しばしばテーブルが現実的でないサイズにまでなる．

本研究では整数値の出力を差分プライバシーの意味で保護することを目的とする．そのために (3) の方針を取り，必要なテーブルのサイズを削減するプロトコルとテーブル作成アルゴリズムを提案する．誤差量は最適な場合に比べて 1.5 から 1.7 倍ほどになるものの，テーブルのサイズを 2.25MiB（平文）にまで縮小することができる．秘密計算のための暗号化に準同型暗号を利用する場合，何個ノイズを作成しようとも通信量は一定である．

[†1] 正確な定義は様々にある．
[†2] 確率質量関数が $\frac{1-p}{1+p}p^{|k|}$ $(p \in (0,1))$ である $\mathbb{Z}$ を台とする離散確率分布．two-sided geometric, symmetric exponential などとも呼ばれる．
[†3] 通常は実数値ノイズ Z について，その絶対値の期待値 $\mathbb{E}[|Z|]$ で定義される．

2. 関連研究

先述の通り，差分プライバシーのためのセキュアサンプリングには大きく分けて 3 種類ある．この節ではそれぞれ具体的な研究を紹介する．

(1) に分類される研究には [3–8] がある．この内 [4,5] は秘密分散上でラプラス分布に従うノイズをサンプルするプロトコルを提案している．その他はベルヌーイ分布に従う乱数から離散ラプラス分布に従うノイズを生成するプロトコルを提案している．これらの手法は実装が難しく計算量が多いのが難点である．

(2) に分類される研究には [9–13] などがある．いずれもラプラス分布に従うノイズを利用する．これらの手法は先述の通りノイズの大きさ（誤差量）が大きくなることが知られており， [14] などで解析されている．

(3) に分類される研究は少ないが Froelicher らの研究 [15] がある．この研究ではラプラス分布に従うノイズを平文で多数サンプルし，暗号化とシャッフルの後に利用している．一回のサンプリングにつき，ノイズのリストは新しく作る．[15] Theorem 1 によれば，(ε, δ) 差分プライバシーを達成するにはテーブルの要素数を $1/\delta$ 以上にする必要がある．通常これは 10^5 以上の大きな値になってしまう．ただし，この研究は実数値を保護しようとしている点で（も）本研究とは問題設定が異なっている．

3. 提案手法：セキュアサンプリングプロトコル

この節ではセキュアサンプリングのための新たなプロトコル Protocol 1 を提案する．新しい点は，事前に準備したテーブルから 2 個以上取り出して利用する点のみである．以下では共通鍵準同型暗号を用いた秘密計算の中で用いる．

Protocol 1 の内，$\#T$ は配列 T の要素数である．

生成されたノイズ Z_{Enc} は秘密計算の参加者 P1 が作成した秘密鍵 sk で保護された秘密計算結果の保護に用いることができる．

3.1 安全性

利用する共通鍵準同型暗号が IND-CPA 安全ならば，Protocol 1 は semi-honest 攻撃者の存在下で安全である．すなわち，Protocol 1 の出力であるノイズ Z_{Enc} について，参加者 P1, P2 は事前に合意した入力 T, n の他に情報を得られない．

実際，参加者 P1 は事前に P2 と合意した入力 T, n の他

Protocol 1: 2 者間セキュアサンプリングプロトコル．参加者は P1 と P2.

Input: 共通（事前に P1, P2 で合意しておく）：テーブル（配列） T, テーブルから取る要素の個数 $n(\geq 1)$.
Output: P1: 無し．P2: P1 の秘密鍵 sk で暗号化されたノイズ Z_{Enc}.

1 **begin**
　/* 前処理 */
2 　P1 は準同型暗号の秘密鍵 sk を生成する．
3 　P1 はテーブル T をシャッフルする．
4 　P1 は T の全要素を秘密鍵 sk で暗号化して T_{Enc} を得る．
5 　P1 は T_{Enc} を P2 に送信する．
　/* 本処理 */
6 　P2 は一様乱数 $i_1, \ldots, i_n \in \{0, \ldots, \#T-1\}$ を選ぶ．
7 　P2 は和 $Z_{\mathrm{Enc}} = \sum_{k=1}^{n} T_{\mathrm{Enc}}[i_k]$ を計算する．
8 　**return** Z_{Enc}
9 **end**

にはなんの情報も得られない．したがって P1 が攻撃者と結託しても問題ない．

また，参加者 P2 は合意済みの入力 T, n の他には暗号化された T_{Enc} が得られるが，T_{Enc} はシャッフルされてから IND-CPA 安全な準同型暗号で保護されている．そのため参加者 P2 の view は，P1 とは独立に P1 が行う処理（秘密鍵生成，シャッフル，暗号化）を行えば，P2 の入力のみからシミュレートすることができる．したがって P2 が攻撃者と結託しても問題ない．

3.2 提案プロトコルの変形

ノイズを複数個得たければ参加者 P2 が行 6–7 を繰り返せば良い．

同様のプロトコルを秘密分散で行う場合，シャッフルに用いる置換が参加者 P2 に知られないようにしなくてはならない．これには例えば Chase ら [16] が提案している secret shuffle が利用できる．

4. 提案手法：テーブル生成アルゴリズム

この節では Protocol 1 で利用できるテーブルを生成するアルゴリズムのを提案し，このアルゴリズムを用いれば秘密計算の出力を差分プライバシーの意味で保護できることを述べる．

最初に記号を以下のように定める．

- 配列のインデックスは 0 から始まるものとする (zero-based).
- 保護したい出力 $q(D)(\in \mathbb{Z})$ の敏感度 (sensitivity) を

$$\Delta = \max_{\substack{D_0, D_1 \in \mathcal{D} \\ D_0, D_1 \text{は隣接している}}} |q(D_0) - q(D_1)|$$

とする．

- 要素数 L の配列 A と非負整数 $n(>0)$ に対して，A の n 重畳み込み A^{*n} を以下で定める．

$$A^{*n}[i] = \sum_{\alpha} \prod_{k=1}^{n} A[\alpha_k] \quad (i = 0, \ldots, nL)$$

和 $\sum_\alpha$ は非負整数の n 個組 $\alpha = (\alpha_1, \ldots, \alpha_n)$ で，$\sum_{k=1}^{n} \alpha_k = i$ を満たすもの全てを渡る和である．$A = A^{*1}$ に注意．

- 確率質量関数 $f_P(k)$ に従う i.i.d. 確率変数 $X_1, \ldots, X_n$ をとったとき，和 $X_1 + \cdots + X_n$ が従う確率分布の確率質量関数を $f_P^{*n}(k)$ とする．$f_P(k)$ が有限台ならば，f_P^{*n} は配列の n 重畳み込みとみなすことができる．

4.1 概要

提案アルゴリズムは，n 重自己畳込み f_P^{*n} が離散ラプラス分布に近い確率分布を作成することを目的に構成された．離散ラプラス分布は整数値クエリに用いることができるノイズの分布として，有用性の点で最適であることが知られている [17].

f_P^{*n} が離散ラプラス分布の確率密度関数に等しくなる f_P は特性関数などを通じて計算することができる [18]．負の二項分布を用いてそのような f_P からサンプルすることもできる [2]．しかしこれらはそれぞれの確率分布から正確にサンプルできる場合に可能な方法である．

本研究では f_P からのサンプリングは事前に用意されたテーブルからランダムに一つ取り出すことで行われる．この場合，確率質量関数 $f_P(k)$ は $\frac{k \text{ により変わる非負整数}}{k \text{ によらない正整数}}$ の形しか取りえない．また，テーブルの要素数は有限であるため，$f_P(k) \neq 0$ となる k は有限個になる．

加えて f_P^{*n} からサンプルされたノイズを用いるプロトコル 1 が差分プライバシーの達成に利用できるためには，次式が成り立つ必要がある（詳細は 4.3 節と 5 節冒頭）．

$$\forall k \in \mathbb{Z},\ k < 0 \text{ and } f_P^{*n}(k) \neq 0 \implies f_P^{*n}(k+1) \leq \exp(\varepsilon/\Delta) f_P^{*n}(k) \quad (1)$$

ただし，f_P が 0 について対称 (i.e. $f_P(k) = f_P(-k)$) だとした．この対称性の仮定は差分プライバシーの定義 1.1 に現れるデータベース D_0, D_1 を対称に扱うことに相当する．

そこで本研究では確率質量関数 $f_P(k)(= f_P(-k))$ を両裾から（0 から遠い k についての値 $f_P(k)$ から）順に決定していくアルゴリズムを構成した．正確には正整数の配列 D がアルゴリズムにより構成され，確率質量関数は $k \mapsto D[|\#D - k|]$ に比例するものとして定まる．両裾から順に $f_P(k)$ の値を決定していくことで，全ての負の整数 k についての条件 (1) が成り立つことが保証する，という大きな問題を，特定の k について不等式が成り立つことを保証する，という小さな問題にしている．

4.2 提案アルゴリズムの詳細

この節では Protocol 1 で利用できるテーブルを生成するアルゴリズムの詳細を述べる．

Algorithm 2 に本論文の提案アルゴリズムを提示する．Algorithm 2 の入力 n は Protocol 1 の入力 n と同じものである．

Algorithm 2: テーブル生成アルゴリズム

Input: プライバシーバジェット $\varepsilon(>0), \delta(\in(0,1/2))$，敏感度 Δ，テーブルから取る要素の個数 $n(\geq 1)$，テーブルの初期値 $\texttt{init}(>0)$

Output: Protocol 1 で利用できるテーブル T

1 **begin**
2 $D \leftarrow [\texttt{init}]$
3 $L \leftarrow 0$
4 **repeat**
5 $D_{\mathrm{sym}}(x) \leftarrow D +\!\!+ [x] +\!\!+ \texttt{reverse}(D)$
6 $s \leftarrow \texttt{solve}((D_{\mathrm{sym}}(x))^{*n}[L+1] == \exp(\varepsilon/\Delta)(D_{\mathrm{sym}}(x))^{*n}[L])$ // x の 1 次方程式の解
7 $D \leftarrow D +\!\!+ [\lfloor s \rfloor]$
8 $D_{\mathrm{sym}} \leftarrow D_{\mathrm{sym}}(x)$
9 $L \leftarrow L+1$
/* D の長さは L，D_{sym} の長さは $2L-1$ */
10 **until** $(L+1 \geq \Delta)$ *and* $\left(\frac{\sum_{k=0}^{\Delta-1} D^{*n}_{\mathrm{sym}}[k]}{\sum_{k=0}^{2L} D^{*n}_{\mathrm{sym}}[k]} \leq \delta\right)$
/* 以上で確率質量関数 $f_P(k) = \frac{D_{\mathrm{sym}}[k]}{\sum_k D_{\mathrm{sym}}[k]}$ が定まる */
11
/* 作成した D からテーブル T を以下で生成する */
12 **for** $i = 0$ **to** $L-1$ **do**
13 $T \leftarrow T +\!\!+ \texttt{repeat}(L-i, D[i])$
14 $T \leftarrow T +\!\!+ \texttt{repeat}(-(L-i), D[i])$
15 **end**
16 $T \leftarrow T +\!\!+ \texttt{repeat}(0, D[i])$
17 **return** T
18 **end**

このアルゴリズム中に現れるサブルーチンは以下のとおりである．

$\texttt{reverse}(A)$　配列 A を反転させる関数．

$\texttt{solve}(f == g)$　1 変数 1 次方程式 $f(x) == g(x)$ を x について解き，解を返す関数．

$\texttt{repeat}(v, n)$　値 v のみが n 個入った配列を返す関数．

また，$D_{\mathrm{sym}}(x)$ は配列を返す関数である．その返値である配列の中央の要素は x である．

4.3 出力の性質

Algorithm 2 の出力 T を Protocol 1 に利用することで，秘密計算の結果を差分プライバシーの意味で保護できる，ということが実験的に確認されている．以下でその性質を述べていく．最初は提案アルゴリズムに限らず成立する命題から始める．

データベース $D \in \mathcal{D}$ についての整数値クエリ $q(D) \in Z$ を次のようなランダム写像（メカニズム）で保護することを考える．

$$\mathcal{M}(D) = q(D) + Z_1 + \cdots + Z_n, \quad Z_1, \ldots, Z_n \sim f_P \tag{2}$$

f_P は整数上の確率分布 P の確率質量関数である．

定理 4.1

生成された確率質量関数 $f_P(k)$ に従う iid 確率変数 $X_1, \ldots, X_n$ をとる．このとき，和 $X_1 + \cdots + X_n$ が従う確率分布 $f^{*n}_P(k)$ が以下の条件を満たすとき，式 (2) のメカニズム $\mathcal{M}$ は (ε, δ) 差分プライバシーを与える．

条件

(i) 任意の $k \in \mathbb{Z}$ について $f^{*n}_P(k) = f^{*n}_P(-k)$.

(ii) ある整数 $w(>1)$ が存在し，整数 $k \in \mathbb{Z}$ が $|k| \leq w$ を満たすならば $f^{*n}_P(k) > 0$ であり，それ以外の k では $f^{*n}_P(k) = 0$ である．

(iii) $-w \leq k < 0$ について $f^{*n}_P(k) < f^{*n}_P(k+1)$.

(iv) $-w \leq k < 0$ について $f^{*n}_P(k+1) \leq \exp(\varepsilon/\Delta) f^{*n}_P(k)$.

(v) $\sum_{-w \leq k < -w+\Delta} f^{*n}_P(k) \leq \delta$.

□

ここからは提案アルゴリズムについて述べる．Algorithm 2 の出力をテーブル T とする．T から一様ランダムに要素を選んだときの，その選ばれた要素の確率分布を P_{Alg2} とし，確率質量関数を $f_{P_{\mathrm{Alg2}}}$ とする．$f_{P_{\mathrm{Alg2}}}$ は Algorithm 2 中の配列 D_{sym} を用いて以下のように表せる．ここでは D_{sym} の要素数を L とする．

$$f_{P_{\mathrm{Alg2}}}(k) = \begin{cases} \frac{D_{\mathrm{sym}}[k]}{\sum_{-L \leq k \leq L} D_{\sum}[i]} & (-L \leq k \leq L) \\ 0 & (\text{otherwise}). \end{cases} \tag{3}$$

上記の条件の内，(i), (ii), (v) が $f_{P_{\mathrm{Alg2}}}, f^{*n}_{P_{\mathrm{Alg2}}}$ について成り立つことは Algorithm 2 の内容から明らかである．十分に初期値 $\texttt{init}$ が大きければ $f^{*n}_{P_{\mathrm{Alg2}}}$ は条件 (iii) を満たす．これは次節で述べる．しかし本稿執筆時点で，一般の n については条件 (iv) が満たされることを証明できていない．

定理 4.2

Algorithm 2 に入力された初期値 $\texttt{init}$ が十分大きいとする．$n = 1$ のとき，$f_{P_{\mathrm{Alg2}}}$ は条件 (iv) を満たす．□

(証明)．Algorithm 2 の行 6, 7 から明らか． ∎

予想 4.2.1

定理 4.2 と同じ前提とする．$n > 1$ のときにも，$f_{P_{\mathrm{Alg2}}}$ は条件 (iv) を満たす． □

しかしこの予想 4.2.1 が成り立つことは，節 5 の通り実験的に確認されている．また，Protocol 1 を実行する前にテーブル T は公開されるため，実際に Algorithm 2 の出力を利用する前に条件 (iv) が満たされることは確かめることができる．したがって上記の予想が証明できていない現時点でも，実用上の問題は事前に避けることができる．

提案アルゴリズムで作成されるテーブルの要素数は次のように上から評価することができる．

命題 4.3

Algorithm 2 が出力するテーブル T を考える．T に対応する確率密度関数 $f_{P_{\mathrm{Alg2}}}$ が条件 (i)–(iv) を満たすとき，テーブル T の要素数 $\#T$ は

$$\#T \leq \left(\frac{\sum_{i=0}^{\Delta-1} \exp(\frac{\varepsilon}{\Delta} i)}{\delta} \right)^{1/n} \cdot \texttt{init}$$

と評価できる． □

4.4 初期値の選び方

初期値 `init` が小さいと，D の要素に負の値が含まれてしまったり，f_P^{*n} が途中から長い区間に渡って一様になってしまったりと，適切な確率質量関数 f_P が得られないことがある．しかし命題 4.3 のとおり `init` は小さいほどテーブルサイズが小さくなり，好ましい．

この問題を避けるためには数列 $D_{\mathrm{sym}}^{*n}[0], \ldots, D_{\mathrm{sym}}^{*n}[L]$ が狭義単調増加するように初期値 `init` を設定すれば十分である．この条件の必要条件は次のようになる．

$$\begin{aligned} & D_{\mathrm{sym}}^{*n}[0] < \cdots < D_{\mathrm{sym}}^{*n}[L] \\ \Longrightarrow\ & D_{\mathrm{sym}}^{*n}[0] < D_{\mathrm{sym}}^{*n}[1] \\ \Longleftrightarrow\ & \left\lfloor \exp(\varepsilon/\Delta) \frac{\texttt{init}}{n} \right\rfloor > \frac{\texttt{init}}{n} \end{aligned} \tag{4}$$

したがって初期値 `init` は少なくとも式 (4) を満たす必要がある．実験的には，命題「I 以上の任意の整数を `init` とすれば式 (4) が成り立つ」を成り立たせる I を初期値 `init` として選べば十分である．確実に保証するためには $D_{\mathrm{sym}}^{*n}[0] < \cdots < D_{\mathrm{sym}}^{*n}[L]$ が成り立つことを Algorithm 2 の 8 行目直後に確認すれば良い．

4.5 関連研究

提案アルゴリズムのように，自己畳込み f^{*2} が所望の分布となるような関数 f を計算することは逆自己畳み込み (inverse autoconvolution) と呼ばれる．信号処理，画像処理，組み合わせ最適化の文脈で問題となるようである．

逆自己畳み込みを扱う研究としては [19–22] がある．このうち最初の 3 つは最適化のアプローチをとっており，その目的関数は最小二乗誤差のような，本研究の目的にはそぐわないものである．フーリエ変換を用いている重弘，山村の研究 [22] で提案されている手法については後の節で実験，比較する．

5. テーブル生成アルゴリズムの実験

節 4 で触れたとおり，テーブル生成アルゴリズムは確率質量関数 f_P を作成する部分と，f_P からテーブル T を作成する部分に分けることができる．テーブル T から一様ランダムに要素を選んだときの，その選ばれた要素の分布の確率質量関数が f_P となるように T は作成される．

この節では既存手法と提案アルゴリズムについて，次で定義される関数 $E[f_P^{*n}]$ を観察する．

$$\mathbb{Z} \ni k \mapsto E[f_P^{*n}](k) = \ln(f_P^{*n}(k+1)) - \ln(f_P^{*n}(k))$$

なお，正でない実数 x について $\ln(x)$ は無限大 ∞ であるとする．f_P から生成されたテーブルを提案プロトコル Protocol 1 で利用したときに (ε, δ) 差分プライバシーが達成されるための必要条件として節 4.3 で述べた条件 (iv) があったが，この条件と $E[f_P^{*n}](k)$ の値域が $[-\varepsilon/\Delta, \varepsilon/\Delta] \cup \{\pm\infty\}$ に含まれることとは同値である．

5.1 既存手法の実験

本研究では提案手法と 2 つの手法を比較する．

4.1 節で述べたとおり，本提案アルゴリズムで生成される確率分布は打ち切り離散ラプラス分布 (TLap) の近似と見ることができる．そこで比較として打ち切りラプラス分布を近似することができる既存手法について実験する．

(A) 打ち切り離散ラプラス分布を定数 A 倍し，整数に丸める手法．定数 A は $\lceil \frac{1}{\text{TLap の正の最小値}} \rceil$．$n = 1$ の場合のみ．

(B) 打ち切り離散ラプラス分布を [22] の手法で逆自己畳込みし，得られた分布を定数 B 倍して整数に丸める手法．定数 B の定め方は手法 (A) と同様．$n = 2$ の場合のみ．

どちらの手法についても，ε は 1.0，敏感度 Δ も 1 とし，δ は 10^{-10} 程度になるように生成パラメータ（アルゴリズムへの入力）を調整して実験した．

それぞれの場合の $E[f_P^{*n}]$ は図 1 と図 2 の様になった．見ての通り $E[f_P^{*n}]$ の値が $[-\varepsilon/\Delta, \varepsilon/\Delta] \cup \{\pm\infty\}$ に含まれていない．したがって，これら既存手法で生成したテーブルを利用すると差分プライバシーを達成することは出来ない．

5.2 提案アルゴリズムの実験

最初に既存手法と同じく，$n = 1, 2$ の場合について提案アルゴリズムが作成する確率質量関数 f_P についての

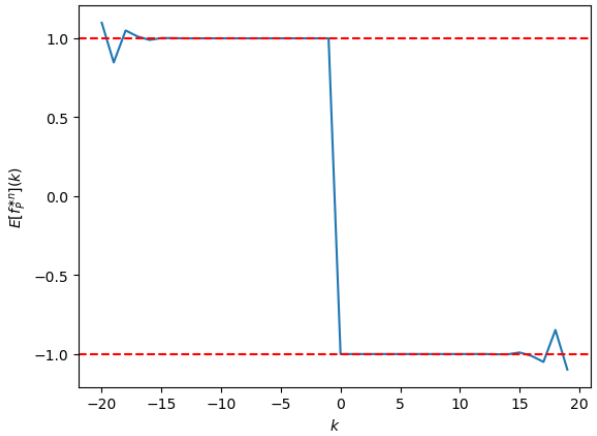

図 1 既存手法 (A) が作成した確率分布 f_P の $E[f_P^{*n}]$. 生成パラメータは $\varepsilon = 1.0, \Delta = 1$, $\underline{n = 1}$. $\delta = 9.52 \times 10^{-10}$ となった. 値域が $[-\varepsilon/\Delta, \varepsilon/\Delta] \cup \{\pm\infty\}$ に**含まれていない**ことが見て取れる. なお, テーブルの要素数は 1049874877, 1 要素 64bit とすると 250GiB 必要になる.

Fig. 1 $E[f_P^{*n}]$ for p.m.f. f_P generated by the exsiting method (A).

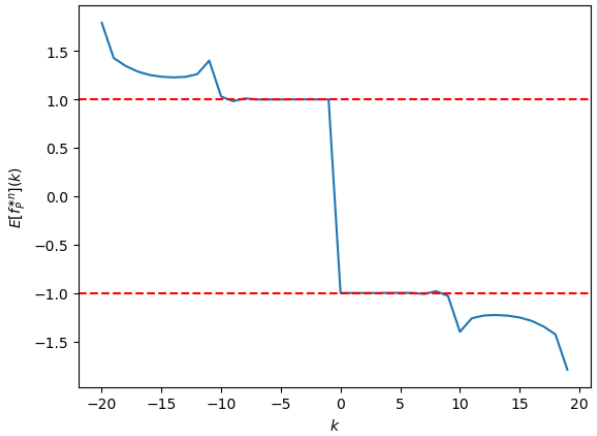

図 2 既存手法 (B) が作成した確率分布 f_P についての $E[f_P^{*n}]$. 生成パラメータは $\varepsilon = 1.0, \Delta = 1$, $\underline{n = 2}$. $\delta = 2.94 \times 10^{-11}$ となった. 値域が $[-\varepsilon/\Delta, \varepsilon/\Delta] \cup \{\pm\infty\}$ に**含まれていない**ことが見て取れる. なお, テーブルの要素数は 184540, 1 要素 64bit とすると 45MiB 必要になる.

Fig. 2 $E[f_P^{*n}]$ for p.m.f. f_P generated by the exsiting method (B).

$E[f_P^{*n}]$ を観察する. 図 3 にそのプロットを示す. 見ての通り $E[f_P^{*n}]$ の値が $[-\varepsilon/\Delta, \varepsilon/\Delta] \cup \{\pm\infty\}$ に含まれている. これが予想 4.2.1 の実験的証拠である. なお, 同様の事実は本論文に記載した全ての入力で確認している.

n と δ, n と ε を同時に変化させたときのテーブルの要素数をそれぞれ表 1 と表 2 に示した. 詳細な条件はそれぞれの表のキャプションに示している. δ あるいは ε が小さくなるにつれてテーブルの要素数が増え, n が大きくなるに連れてテーブルの要素数が小さくなっていることが観察できる. 表 1 をさらに詳細に観察すれば, テーブルの要素数

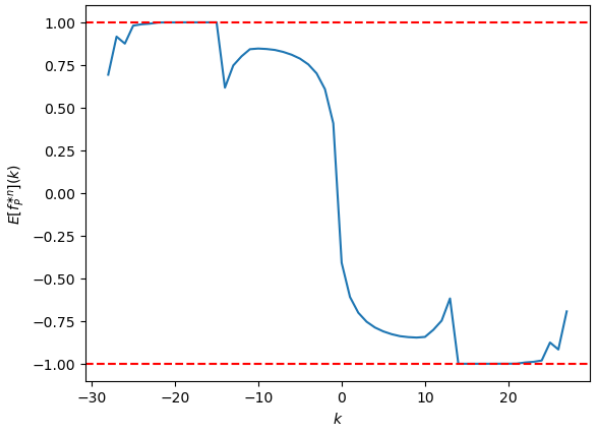

図 3 提案手法 2 が生成した確率分布 f_P と f_P^{*n} 及び $E[f_P^{*n}]$. 生成パラメータは $\varepsilon = 1.0, \Delta = 1, \delta \leq 10^{-10}$, $\underline{n = 2}$. $\delta = 1.15 \times 10^{-11}$ となった. 値域が $[-\varepsilon/\Delta, \varepsilon/\Delta] \cup \{\pm\infty\}$ に**含まれている**ことが見て取れる. なお, テーブルの要素数は 295384, 1 要素 64bit とすると 2.25MiB 必要になる.

はおおよそ $(1/\delta)^{1/n}$ のように増減することが分かる. これは命題 4.3 が示すとおりである.

δ \ n	1	2	3	4
10^{-4}	30641	149	146	42
10^{-6}	1662884	2454	357	97
10^{-8}	246792753	16505	2256	583
10^{-10}	36627290627 (273GiB)	295384 (2.25MiB)	14731 (115KiB)	1466 (11.4KiB)

表 1 加算回数 n, プライバシーパラメータ δ と Algorithm 2 で生成されるテーブルのサイズの関係. プライバシーパラメータ $\varepsilon = 1.0$, 敏感度 $\Delta = 1$ は固定. 最下行にはテーブルの各要素を 64bit の整数で表現したときに必要な容量を括弧内に示した.

ε \ n	1	2	3	4
1.0	1662884	2454	357	97
0.5	3278624	6218	963	365
0.25	8224233	15452	1983	891
0.1	20537623 (156MiB)	39740 (310KiB)	5483 (42.8KiB)	2391 (18.7KiB)

表 2 加算回数 n, プライバシーパラメータ ε と Algorithm 2 で生成されるテーブルのサイズの関係. プライバシーバパラメータ $\delta = 10^{-6}$, 敏感度 $\Delta = 1$ は固定. 最下行にはテーブルの各要素を 64bit の整数で表現したときに必要な容量を括弧内に示した.

実験の最後として, 表 3 に ε と n を変化させたときの誤差量, すなわち Z を $f_{P_{\mathrm{Alg2}}}^{*n}$ に従う確率変数とするときの $\mathbb{E}[|Z|]$ を示す. プライバシーバパラメータ $\delta = 10^{-6}$, 敏感度 $\Delta = 1$ は固定した. この値は正確な離散ラプラス分布で

は $\Delta/\varepsilon(=1/\varepsilon)$ となる．$n=1$ のときには打ち切っている影響で誤差量は $1/\varepsilon$ より小さくなるが，n が大きくなるに連れて誤差量が増えていく様子が観察できる．そのため，実用上は $n=2,3$ 程度が良いと考えられる．

ε \ n	1	2	3	4
1.0	0.852	1.482	2.119	2.923
0.5	1.919	3.197	4.456	5.953
0.25	3.959	6.454	9.268	12.187
0.1	9.986	16.648	23.816	31.365

表 3 Algorithm 2 で生成される確率質量関数を $f_{P_{\mathrm{Alg2}}}(k)$ としたときの，加算回数 n，プライバシーパラメータ ε と ℓ_1 誤差 $\sum_{k\in\mathbb{Z}}|k|f^{*n}_{P_{\mathrm{Alg2}}}(k)$ の関係．正確な離散ラプラス分布では Δ/ε となる．プライバシーバパラメータ $\delta=10^{-6}$，敏感度 $\Delta=1$ は固定．

6. 結論

本研究では暗号化したテーブルから $n(>1)$ 回引き出し，その和をノイズとして利用するプロトコルを提案した（3 節）．同時に，このプロトコルで得られるノイズが差分プライバシーの意味での保護に利用できるようにするため，テーブルの生成アルゴリズムを提案した（4 節）．また，生成されるテーブルと確率質量関数の性質を理論と実験により示した（4.3 節，5 節）．

実験で観察したとおり，n が大きくなるに連れてテーブルの要素数はおおよそ $(1/\delta)^{1/n}$ に比例するように小さくなるが，誤差量 $\mathbb{E}[|Z|]$ は大きくなっていく（5 節）．そのため，実用上は $n=2,3$ 程度が良いと考えられる．

6.1 残された課題

残った課題は 2 つある．一つは n が大きくても誤差量が小さいような分布を生成するアルゴリズムの構築である．もう一つは予想 4.2.1 の証明である．どちらの解決にも，本研究の提案アルゴリズムが生成する確率質量関数 $f_{P_{\mathrm{Alg2}}}$ を理論的に深く知ることが必要だと思われる．まだわかっていないこととして，例えば図 4 から分かるように，$n>1$ のとき $f_{P_{\mathrm{Alg2}}}$ の値は A 字形ではなく W 字になる．4.1 節のように n 重自己畳込み f_P^{*n} が正確に離散ラプラス分布になるような f_P ではこのような様子は見られない．

謝辞 提案プロトコルの安全性（3.1 節）について，NTT 社会情報研究所属の市川 敦謙さんにコメント頂きました．ありがとうございました．

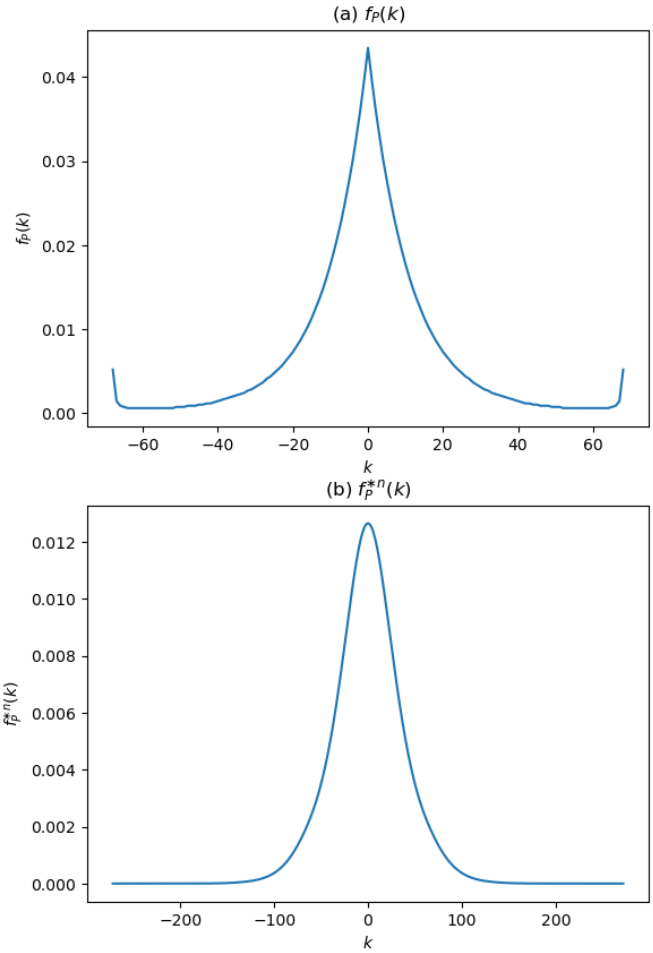

図 4 $\varepsilon=0.1, n=4$ のときに Algorithm 2 が作成する確率質量関数 $f_{P_{\mathrm{Alg2}}}$ と $f^{*n}_{P_{\mathrm{Alg2}}}$．その他の入力は $\delta=7.08\times10^{-10}(\leq10^{-8}), \Delta=1$．$f_{P_{\mathrm{Alg2}}}$ の両裾が上がり，W 字型になっている．上がり方は小さくなるが，$n=2$ でもこのように両裾が上がる．

参考文献

[1] 佐久間 淳：データ解析におけるプライバシー保護，講談社 (2016).
[2] Inusah, S. and Kozubowski, T. J.: A Discrete Analogue of the Laplace Distribution, *Journal of Statistical Planning and Inference*, Vol. 136, No. 3, pp. 1090–1102 (online), DOI: 10.1016/j.jspi.2004.08.014 (2006).
[3] Dwork, C., Kenthapadi, K., McSherry, F., Mironov, I. and Naor, M.: Our Data, Ourselves: Privacy Via Distributed Noise Generation, *Advances in Cryptology - EUROCRYPT 2006* (Vaudenay, S., ed.), Lecture Notes in Computer Science, Berlin, Heidelberg, Springer, pp. 486–503 (online), DOI: 10.1007/11761679_29 (2006).
[4] Eigner, F., Kate, A., Maffei, M., Pampaloni, F. and Pryvalov, I.: Differentially Private Data Aggregation with Optimal Utility, *Proceedings of the 30th Annual Computer Security Applications Conference*, ACSAC '14, New York, NY, USA, Association for Computing Machinery, pp. 316–325 (online), DOI: 10.1145/2664243.2664263 (12 月 8, 2014).
[5] Wu, G., He, Y., Wu, J. and Xia, X.: Inherit Differential Privacy in Distributed Setting: Multiparty Randomized Function Computation, *2016 IEEE Trustcom/BigDataSE/ISPA*, pp. 921–928 (online), DOI: 10.1109/TrustCom.2016.0157 (2016).
[6] Champion, J., shelat, a. and Ullman, J.: Securely Sampling Biased Coins with Applications to Differential Privacy, *Proceedings of the 2019 ACM SIGSAC Confer-*

ence on Computer and Communications Security, London United Kingdom, ACM, pp. 603–614 (online), DOI: 10.1145/3319535.3354256 (2019).

[7] 江利口 礼央，市川 敦謙，國廣 昇：秘密計算への応用に向けた差分プライバシを達成する効率的なノイズ生成，SCIS 2020 予稿集，Kochi, Japan (Jan. 28 – 31, 2020).

[8] 紀伊 真昇，市川 敦謙，千田 浩司，濱田 浩気：差分プライベートな秘密計算のための暗号化された離散乱数を生成する非対話型二者間プロトコル，第 186 回マルチメディア通信と分散処理・第 92 回コンピュータセキュリティ合同研究発表会 (2021).

[9] Pathak, M., Rane, S. and Raj, B.: Multiparty Differential Privacy via Aggregation of Locally Trained Classifiers, *Advances in Neural Information Processing Systems*, Vol. 23, Curran Associates, Inc. (2010).

[10] Clifton, C. and Anandan, B.: Challenges and Opportunities for Security with Differential Privacy, *Information Systems Security* (Bagchi, A. and Ray, I., eds.), Lecture Notes in Computer Science, Berlin, Heidelberg, Springer, pp. 1–13 (online), DOI: 10.1007/978-3-642-45204-8_1 (2013).

[11] Goryczka, S., Xiong, L. and Sunderam, V.: Secure Multiparty Aggregation with Differential Privacy: A Comparative Study, *Proceedings of the Joint EDBT/ICDT 2013 Workshops*, EDBT '13, New York, NY, USA, Association for Computing Machinery, pp. 155–163 (online), DOI: 10.1145/2457317.2457343 (3 月 18, 2013).

[12] Chowdhury, A. R., Wang, C., He, X., Machanavajjhala, A. and Jha, S.: Cryptϵ: Crypto-Assisted Differential Privacy on Untrusted Servers, *arXiv:1902.07756 [cs]* (2020).

[13] Acar, A., Celik, Z. B., Aksu, H., Uluagac, A. S. and McDaniel, P.: Achieving Secure and Differentially Private Computations in Multiparty Settings, *2017 IEEE Symposium on Privacy-Aware Computing (PAC)*, pp. 49–59 (online), DOI: 10.1109/PAC.2017.12 (2017).

[14] Chan, T.-H. H., Shi, E. and Song, D.: Optimal Lower Bound for Differentially Private Multi-party Aggregation, *Algorithms – ESA 2012* (Epstein, L. and Ferragina, P., eds.), Lecture Notes in Computer Science, Berlin, Heidelberg, Springer, pp. 277–288 (online), DOI: 10.1007/978-3-642-33090-2_25 (2012).

[15] Froelicher, D., Egger, P., Sousa, J. S., Raisaro, J. L., Huang, Z., Mouchet, C., Ford, B. and Hubaux, J.-P.: UnLynx: A Decentralized System for Privacy-Conscious Data Sharing, *Proceedings on Privacy Enhancing Technologies*, Vol. 2017, No. 4, pp. 232–250 (online), DOI: 10.1515/popets-2017-0047 (2017).

[16] Chase, M., Ghosh, E. and Poburinnaya, O.: Secret Shared Shuffle, Technical Report 1340 (2019).

[17] Ghosh, A., Roughgarden, T. and Sundararajan, M.: Universally Utility-Maximizing Privacy Mechanisms, *Proceedings of the Forty-First Annual ACM Symposium on Theory of Computing*, STOC '09, New York, NY, USA, Association for Computing Machinery, pp. 351–360 (online), DOI: 10.1145/1536414.1536464 (5 月 31, 2009).

[18] Seetha Lekshmi, V and Simi Sebastian: A Skewed Generalized Discrete Laplace Distribution, Vol. 2, No. 3, p. 8 (2014).

[19] Martinez, V.: Global Methods in the Inversion of a Self-Convolution, *Journal of Electron Spectroscopy and Related Phenomena*, Vol. 17, No. 1, pp. 33–43 (online), DOI: 10.1016/0368-2048(79)85025-2 (1979).

[20] Dose, V., Fauster, T. and Gossmann, H. J.: The Inversion of Autoconvolution Integrals, *Journal of Computational Physics*, Vol. 41, No. 1, pp. 34–50 (online), DOI: 10.1016/0021-9991(81)90078-4 (5 月 1, 1981).

[21] Finesso, L. and Spreij, P.: The Inverse Problem of Positive Autoconvolution, *arXiv:2111.14430 [math]* (2021).

[22] 重弘 裕二，山村 真由子：確率分布推定のための逆自己畳み込みアルゴリズム，電気学会論文誌 C（電子・情報・システム部門誌），Vol. 141, No. 7, pp. 802–811（オンライン），DOI: 10.1541/ieejeiss.141.802 (2021).