Vol.2021-DPS-186 No.5
Vol.2021-CSEC-92 No.5
2021/3/15

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

紀伊 真昇[1,a)]　市川 敦謙[1]

概要：秘密計算は参加者がそれぞれの秘密情報を自身以外に漏洩させずに，秘密情報に統計処理などの処理を施すことを可能にする．最近では秘密計算での処理結果にノイズを加えて差分プライバシーを達成する手法が研究されている．そういった手法を実現するには安全な乱数，すなわち，いずれの参加者も事前に定めた分布以上のことは知らない乱数を生成する必要がある．特に離散乱数を安全に生成することは，様々な p についてベルヌーイ分布 $\mathrm{Ber}(p)$ に従う乱数を安全に生成することに帰着される．しかしこれは難しく，従来手法では参加者が三人以上必要，通信量が大きい，といった問題点が有った．本研究では完全準同型暗号方式の一つである TFHE 方式を用いて，確率 $p = \frac{0}{N}, \frac{1}{N}, \frac{2}{N}, \ldots, \frac{N}{N}$（$N$ は TFHE のパラメータ．2^{10} 程度.）についてベルヌーイ分布 $\mathrm{Ber}(p)$ に従う乱数を予め TFHE 方式で暗号化された形で安全に生成する手法を提案する．この手法では参加者は二人いれば十分であり，また準備の他に通信は不要である．論文では応用として，生成した乱数から予め暗号化された形で一様分布，二項分布，離散ラプラス分布に従う離散乱数を生成する手法を述べる．これらの乱数は差分プライバシーを達成するために利用することができる．

A Non-Interactive Two-Party Protocol that Generate Encrypted Discrete Random Numbers for Differential Private Secure Computation

1. はじめに

昨今，企業や病院などの組織が得た個人情報や売上情報などの秘密情報を，安全に活用したいという需要が高まっている．一つの組織内の情報を分析したいという需要もあるが，複数の病院が持つ電子カルテといった異なる組織の秘密情報を組み合わせて分析したいという需要も大きい．

そのため参加者がそれぞれの秘密情報を他者に漏洩させずに，秘密情報に統計処理などの処理を施すことを可能にする秘密計算技術への注目が高まっている．秘密計算技術は秘密分散 (Secret Sharing)，歪曲回路 (Garbled Circuit)，準同型暗号 (Homomorphic Encryption) といった技術で実現される．

しかし秘密計算で処理した結果をそのまま参加者に公開すると，出力結果から各参加者の秘密情報の一部が漏れる危険性がある．そのため秘密計算下で処理結果に手を加えて差分プライバシー [7] を達成する手法が研究されている．準同型暗号と差分プライバシーを組み合わせた手法のサーベイとして，たとえば牛山ら [16] がある．

差分プライバシーを達成する手法としてはラプラスメカニズム [7] など，計算結果に適切なノイズを加えて出力とする手法が主流である．これを秘密計算で行う場合，加えるノイズは安全な乱数，すなわち，いずれの参加者も事前に定めた分布以上のことは知らない乱数でなくてはならない．

本研究では特に離散乱数を安全に生成する手法をあつかう．多くの離散確率分布 D について，分布 D に従う乱数を生成することは，様々な p についてベルヌーイ分布 $\mathrm{Ber}(p)$ に従う乱数を生成することに帰着される．例えば二項分布 $\mathrm{Bin}(m,p)$ に従う乱数を生成するには，ベルヌーイ分布 $\mathrm{Ber}(p)$ に従う乱数を m 個生成し，そのうち何個が 1 なのかを数えれば良い．そのため本研究でもベルヌーイ分布にしたがう乱数を安全に生成する手法を主眼においている．

[1] NTT セキュアプラットフォーム研究所, 〒 180–0012 東京都武蔵野市緑町 3–9–11
NTT Secure Platform Laboratories, 3–9–11 Midori-cho, Musashino-shi, Tokyo 180–0012, Japan.
[a)] masanobu.kii.gw@hco.ntt.co.jp

1.1 先行研究

秘密計算の中で特に離散乱数を生成する方法の先行研究を紹介する．

ベルヌーイ分布 Ber(1/2) に従う乱数は，ブール値の XOR ができる秘密計算方式であれば安全に生成することができる．これには各参加者がベルヌーイ分布 Ber(1/2) に従う乱数を生成して秘匿化し，秘密計算でそれらの XOR を計算すれば良い．先行研究の多くではこうして生成された乱数から目的の分布に従う乱数を生成する．

Dwork らの先行研究 [8] ではベルヌーイ分布 Ber(1/2) に従う $2n\log(n+d)$ 個の乱数を入力すると，ベルヌーイ分布 Ber(p) に従う乱数を n 個生成する回路が提案されている．この [8] で提案されている回路の深さは $\Theta(\log(n+d))$，サイズは $\Theta(nds^2\log(n+d))$ となっている．ここで d は p の精度，s は回路のパラメータで小さな正整数である．

江利口らの先行研究 [17] ではベルヌーイ分布 Ber(1/2) に従う n 個の乱数を n ビットの整数とみなし，この整数と定数との比較結果を以て Ber(p) に従う乱数とする．二進表現された整数同士の秘密計算下での比較は秘密分散などを用いて実現できるが，大小比較の回路のサイズは $O(n)$ ほどあるため，必要な通信量の評価も $O(n)$ となる．

差分プライベートな出力をする分散データベースシステムである UnLynx [10] や Drynx [9] では，乱数を事前に十分大量に生成し，準同型暗号や加法秘密分散で秘匿されたテーブルに格納する．安全に生成できる離散一様乱数をインデックスとしてテーブルを参照し，格納されていた値を乱数として用いる．しかしこの手法では参加者（サーバ）がノイズのテーブルを完全に把握しているため，(ϵ,δ)-差分プライバシーを達成するためには $1/\delta$ 個以上の乱数を事前に準備する必要がある ([10] Theorem 1)．

1.2 本稿の成果

本稿では完全準同型暗号 TFHE が用いるブートストラップアルゴリズムに着目し，以下の特徴を持つ二者間プロトコルを提案する．

- 確率 $p \in \left\{0=\frac{0}{N},\frac{1}{N},\frac{2}{N},\ldots,\frac{N}{N}=1\right\}$ [*1]についてベルヌーイ分布 Ber(p) に従う乱数を生成することができる．
- プロトコルに参加する二者の内，一人の役割は，完全準同型暗号方式 TFHE のブートストラップ鍵を作ってもう一人に送信することだけである．すなわちこのプロトコルは非対話型プロトコルである．
- 行われる通信は上述の鍵の送受信だけである．したがって通信量は作る乱数の個数によらない．
- ベルヌーイ分布に従う一つの乱数を生成するのに必要な計算量は，TFHE のブートストラップに必要な計算量とほぼ同じである．例えば TFHE 方式の実装の一つ NuFHE [14] では 0.13 ミリ秒でブートストラップが実行できる．

さらに，生成したベルヌーイ分布に従う乱数から，一様分布，二項分布，離散ラプラス分布に従う離散乱数を予め暗号化された形で生成する手法を述べる．これらは乱数を二進数表示した際の各桁（0 または 1）が TFHE 方式で暗号化された形で生成される．

[*1] N は TFHE のパラメータ．通常 $2^{10},2^{11},2^{12}$ のいずれかを用いる．

2. 準備: 準同型暗号と差分プライバシー

本研究では準同型暗号を用いて差分プライバシーを達成するための乱数を生成することを目的とする．そのため提案手法を述べる前に準同型暗号と差分プライバシーを紹介する．

2.1 準同型暗号

準同型暗号は，暗号化されたデータに対して復号することなく計算処理を施すことができる暗号である．例えば，秘密情報を信頼できない他者に渡して計算させ出力を受け取る，ということを安全に行うことができるようになる．

準同型暗号 HE は通常の暗号と同じく鍵生成 HE.KeyGen，暗号化 HE.Enc，復号 HE.Dec のアルゴリズムを備えているのに加え，評価 HE.Eval のアルゴリズムを備えている．評価 HE.Eval は n 引数の計算可能な関数 f（通常は論理回路や算術回路として与えられる）と，鍵 k で暗号化された n 個の暗号文 $x_1,\ldots,x_n$，そして評価鍵 evk と呼ばれる情報を入力として与えられると，$\mathtt{HE.Dec}_k(c)=f(x_1,\ldots,x_n)$ を満たす暗号文 c を出力する．評価鍵 evk は具体的な準同型暗号方式によって様々だが，いずれにせよ評価鍵 evk から鍵 k の情報は漏れない．

一般に準同型暗号で評価できる関数（論理回路や算術回路）f には具体的な準同型暗号方式ごとに制約がある．例えば関数 f として和しか使えない方式や，和・積の回数の上限が決められているものもある．こういった制約がなく，任意の計算可能な関数 f について関数 f の評価 HE.Eval ができる準同型暗号方式は完全準同型暗号方式と呼ばれる．

2.2 差分プライバシー

差分プライバシーは Dwork によって [7] で導入された，秘密情報を含むデータベースから計算された出力についての安全性の定義である．データベースの一要素が変わっても出力結果がほとんど変わらなければ，変わった一要素の情報は出力にほとんど含まれていない，すなわち安全だろう，ということを定式化している．計算は一般には確率的なものなので，次のように確率分布の差を用いて定式化されている．

定義 1. ϵ,δ を 0 以上の実数とする．ランダム化関数 (randomized function) K が (ϵ,δ)-差分プライバシーを持つと

は，高々一つのレコードが異なる任意の二つのデータベース D_1, D_2 と，K の値域の任意の部分集合 S について，

$$\Pr[K(D_1) \in S] \leq e^{\epsilon} \Pr[K(D_2) \in S] + \delta$$

を満たすということ．$(\epsilon, 0)$-差分プライバシーは単に ϵ-差分プライバシーと呼ばれる．

ϵ と δ が 0 に近いほど強い安全性と考えられる．

差分プライベートでない関数（アルゴリズム）f を差分プライベートなランダム化関数にする手法は様々あるが，よく用いられるラプラスメカニズムなどの手法では冒頭で述べたように f の出力にノイズを加えることで差分プライバシーを実現する．加えるべきノイズの分布を決めるには，f の出力が入力にどれほど左右されるかを表す「鋭敏度」を知る必要がある．

3. 準備: 完全準同型暗号方式 TFHE

本稿の提案手法は完全準同型暗号の一つである TFHE 方式をもとにしているため，この節で紹介する．紹介は本稿に必要な範囲を出来るだけ簡潔に紹介するにとどめる．レベルごとのパラメータの違いなどを書いていないため，詳しくは TFHE が提案された論文 [6] を参照してほしい．

TFHE は (Ring) LWE 問題から派生した TLWE 問題をベースとする完全準同型暗号である．[4], [5], [6] で提案された．トーラス $\mathbb{T}$（下記）を用いる点と，暗号文が三種類ある点が大きな特徴である．

まず TFHE の中で用いる代数学的概念について記号をまとめて定義する．

- n を十分大きい正の整数とする．また，N は十分大きい 2 の累乗数とする．k は小さい正の整数とする．
- $\mathbb{Z}, \mathbb{R}$ はそれぞれ整数，実数全体の集合を表す．
- $\mathbb{T} = \mathbb{R}/\mathbb{Z}$ は $\mathbb{Z}$ 加群同士の剰余加群である．
- $\mathbb{T}$ の元は実数 $-\frac{1}{2} \leq x \leq \frac{1}{2}$ を用いて $x \in \mathbb{T}$, $x \bmod \mathbb{Z}$ のように表す．$-\frac{1}{2} \bmod \mathbb{Z} = \frac{1}{2} \bmod \mathbb{Z}$ に注意．
- $\mathbb{Z}_N[X] = \mathbb{Z}[X]/(X^N+1)$, $\mathbb{T}_N[X] = \mathbb{T}[X]/(X^N+1) = \mathbb{T} \otimes_{\mathbb{Z}} \mathbb{Z}_N[X]$ とおく．$X^N + 1$ が $2N$ 次円分多項式であることに注意せよ．
- 元 $p \in \mathbb{T}_N[X]$ の代表元のうち，0 次以上 N 次未満の項のみを含むものは一意に定まるので，これを $\mathrm{Rep}(p)$ と表す．
- $\mathbb{R}$ の部分集合 $\mathbb{B}$ を $\mathbb{B} = \{0, 1\}$ とおく．$\mathbb{Z}_N[X]$ の元 p の内，代表元 $\mathrm{Rep}(p)$ の係数が 0 または 1 であるもの全体の集合を $\mathbb{B}_N[X]$ と表す．
- 実数 x に対し，x を偶数丸めしたものを $\lceil x \rfloor$ で表す．
- $\bar{t} = t \bmod \mathbb{Z} \in \mathbb{T}, n \in \mathbb{Z}$ について，$\bar{t}$ の代表元 t をどう取っても $\lceil nt \rfloor \bmod n\mathbb{Z} \in \mathbb{Z}/n\mathbb{Z}$ は一意に定まる．なのでこの元を $\lceil n\bar{t} \rfloor$ と表す．

3.1 TFHE の暗号文

3.1.1 TLWE 型暗号文

三種類ある暗号文の形式のうち，中心となるのは TLWE 型である．鍵，平文，暗号文の空間は次の通り．

鍵 列 $(s_1, \ldots, s_n) \in \mathbb{B}^n$,
平文 $m \in \{0, \frac{1}{2}\} = \frac{1}{2}$,
暗号文 $(a_1, \ldots, a_n, b) \in \mathbb{T}^{n+1}$.

平文 $m \in \{0, \frac{1}{2}\}$ の暗号化の手順は次のようである．まずトーラスの n 個の元 $a_1, \ldots, a_n$ をランダムに選ぶ．さらに代表元がごく小さい実数であるようなトーラスの元 ε（ノイズと呼ばれる）を選ぶ．通常，ノイズの分布は平均が 0 で分散がごく小さい正規分布を $\left[-\frac{1}{2}, \frac{1}{2}\right]$ の範囲で切ったものにする．このとき $b = m + \varepsilon + \sum_{i=1}^{n} s_i a_i$ とすると，m に対応する TLWE 型の暗号文 $(a_1, \ldots, a_n, b)$ が得られる．特に $(\mathbf{0}, m)(= (0, \ldots, 0, m))$ は m の暗号文とみなせる．

復号は $\phi = b - \sum_{i=1}^{n} s_i a_i$ を計算し，$|\phi| < \frac{1}{4}$ ならば $m' = 0$，そうでなければ $m' = \frac{1}{2}$ を復号結果とする．

3.1.2 TRLWE 型暗号文

TRLWE 型暗号文は様々な処理をする際の中間表現として用いられる．鍵，平文，暗号文の空間は次の通り．

鍵 列 $(s_1, \ldots, s_n) \in \mathbb{B}_N[X]^k$,
平文 $m \in \frac{1}{2}\mathbb{B}_N[X]$,
暗号文 $(a_1, \ldots, a_n, b) \in \mathbb{T}_N[X]^{k+1}$.

暗号化と復号の説明は省略する．なお，[6] では TLWE 型暗号文は TRLWE 型暗号文で $N = 1, k = n$ とした特殊な場合として定義されている．

3.1.3 TRGSW 型暗号文

最後の TGSW 型暗号文は $\mathbb{Z}[X]/(X^N+1)$ の元を平文とする．特に 0 または 1 の TGSW 型暗号文を使うことが多い．鍵，平文，暗号文の空間は次の通り．

鍵 列 $(s_1, \ldots, s_n) \in \mathbb{B}_N[X]^k$,
平文 $m \in \frac{1}{2}\mathbb{B}_N[X]$,
暗号文 $\mathbb{T}_N[X]$ 係数の $(k+1)\ell \times (k+1)$ 行列 C.

ℓ は正の整数で，TRGSW 型暗号文特有のパラメータである．

TFHE 方式では TLWE 型暗号文と TGSW 型暗号文の積を計算することができる．積は TRLWE 型暗号文となる．

3.1.4 暗号文についての記号

TFHE の暗号文について紹介できたので，暗号文についての記号も導入する．

- 鍵 k，平文 m に対応する暗号文の集合を，暗号文の型ごとに $\mathrm{TLWE}_k(m)$, $\mathrm{TRLWE}_k(m)$, $\mathrm{TRGSW}_k(m)$ のように表す．鍵 k はしばしば省略する．
- 暗号文 c に対応する平文を $\mathrm{msg}(c)$ と表す．

前者は暗号文を型と平文を明示する際に，後者は型は明らかとして暗号文の平文だけ明示する際に用いる．

3.2 TFHE のアルゴリズム

[6] で述べられているアルゴリズムのうち本稿の説明で必要なものと，Programmable Bootstrapping を説明する．Programmable Bootstrapping は [6] で述べられていないが，Gate Bootstrapping と 本稿の提案プロトコルの共通の部品として切り出したものである．

3.2.1 BlindRotate と SampleExtract

次の二つのアルゴリズム 1, 2 は TFHE [6] の他のアルゴリズムによく現れる，基礎的なものである．入出力だけ示す．

Algorithm 1: `BlindRotate`, [6] §4.3

Input: 暗号文 $c_v \in \mathrm{TRLWE}_k(v)$，元
$\bar{a}_1, \ldots, \bar{a}_n, \bar{b} \in \mathbb{Z}/2N\mathbb{Z}$, n 個の暗号文
$C_i \in \mathrm{TRGSW}_{k_2}(s_i)$.
Output: 暗号文 $c' \in \mathrm{TRLWE}_{k_2}\left(X^{-\bar{\phi}}v\right)$，ただし
$\bar{\phi} = \bar{b} - \sum_{i=1}^{n} s_i \bar{a}_i$.

Algorithm 2: `SampleExtract`, [6] §4.2

Input: 暗号文
$c \in \mathrm{TRLWE}_k\left(\frac{1}{2}(b_0 + b_1 X^1 + \cdots + b_{N-1}X^{N-1})\right)$.
Output: 暗号文 $c_0 \in \mathrm{TLWE}_{k_0}\left(\frac{1}{2}b_0+\right)$.

SampleExtract の出力に現れる鍵 $k_0 = (s_1, \ldots, s_n) \in \mathbb{B}^n$ は鍵 $k = (S_1, \ldots, S_n) \in \mathbb{B}_N[X]^n$ から，s_i は $\mathrm{Rep}(S_i)$ の定数項，という形で得られる．

3.2.2 Programmable Bootstrapping

Programmable Bootstrap は写像 $f\colon \mathbb{Z}/2N\mathbb{Z} \to \mathbb{T}$ の評価を暗号化した状態で行うアルゴリズムである．つまり $m \in \mathbb{T}$ の暗号文から $f(m+\epsilon)$ (ϵ は小さなノイズ) の暗号文を得ることができる．このアルゴリズム `ProgBootstrap` のアイデアは [2] にある．

Algorithm 3: `ProgBootstrap`

Input: $f(i-N) = f(i)$ $(i \in \mathbb{Z}/2N\mathbb{Z})$ を満たす
写像 $f\colon \mathbb{Z}/2N\mathbb{Z} \to \mathbb{T}$,
暗号文 $c = (a_1, \ldots, a_n, b) \in \mathrm{TLWE}_{k_0=(s_1,\ldots,s_n)}(m)$,
n 個の暗号文
$\mathrm{BK} = (\mathrm{BK}_1, \ldots, \mathrm{BK}_n)$ $(\mathrm{BK}_i \in \mathrm{TRGSW}_{k_2}(s_i))$.
Output: 暗号文 $c'' \in \mathrm{TRLWE}_{k_2}(f(\lceil 2Nm \rfloor + \varepsilon))$．ただし
ε は小さなノイズ．

begin
  $v \leftarrow \sum_{i=0}^{N-1} f(i)X^i$
  $\bar{a}_i \leftarrow \lceil 2Na_i \rfloor$ $(i = 1, \ldots, n), \bar{b} \leftarrow \lceil 2Nb \rfloor$
  $c' \leftarrow$ `BlindRotate` $\left((\mathbf{0}, v), (\bar{a}_1, \ldots, \bar{a}_n, \bar{b}), \mathrm{BK}\right)$
  // $c' \in \mathrm{TRLWE}(X^{\bar{\phi}}v), \bar{\phi} = \lceil 2Nm + \varepsilon \rfloor$.
  $c'' \leftarrow$ `SampleExtract`(c').
  return c''
end

ノイズ ε 大きさの見積もりは [6] Theorem 6.2 （の証明）に述べられている．通常は入力と出力の暗号文で使われている鍵が異なるため，[6] §4.1 の “Key Switching” を用いて入力と出力の鍵を同じものにする．

このアルゴリズム `ProgBootstrap` のアイデアは [2] にあるが，[2] にある正当性の証明は厳密でない．そのためこのアルゴリズムの正当性の証明を 4.1 節で行う．

3.2.3 Gate Bootstrapping

Gate Bootstrapping は本稿の提案手法には現れないが，TFHE の重要なアルゴリズムであるため，また前節のアルゴリズム `ProgBootstrap` の理解の助けとするために紹介する．

Gate Bootstrapping は大きいノイズ（$\frac{1}{8}$ 以下）をもつ TLWE 型暗号文を，平文が同じで小さなノイズをもつ TLWE 型暗号文に変換するアルゴリズムである．一つの論理ゲートを実行するごとにこの Bootstrap をする必要があるのでこのように名前がついている．[6] では Gate Bootstrapping よりも大幅にノイズを削減するアルゴリズム Circuit Bootstrapping が提案されている．

`Bootstrap` では次の写像を用いる．ここで $\mu \in \mathbb{T}$ は任意の値であり，$\mu' \in \mathbb{T}$ は $2\mu' = \mu$ を満たす元（二つある内の一つ）である．

$$f_{\texttt{Bootstrap}}(i) = \begin{cases} \mu' \bmod \mathbb{Z} & (N/2 \le i \le 3N/2) \\ -\mu' \bmod \mathbb{Z} & (\text{上記以外の場合}) \end{cases}$$

この写像が上述のアルゴリズム `ProgBootstrap` の入力に課せられた条件 $f_{\texttt{Bootstrap}}(i-N) = -f_{\texttt{Bootstrap}}(i)$ を満たすことは明らかである．

Algorithm 4: `Bootstrap`, [6] Algorithm 9

Input: 定数 $\mu \in \mathbb{T}$，暗号文 $c = (a_1, \ldots, a_n, b)$
$\in \mathrm{TLWE}_{k_0=(s_1,\ldots,s_n)}(m)$，$n$ 個の暗号文
$\mathrm{BK} = (\mathrm{BK}_1, \ldots, \mathrm{BK}_n)$ $(\mathrm{BK}_i \in \mathrm{TRGSW}_{k_2}(s_i))$.
Output: $|\mathrm{msg}(c)| < \frac{1}{4}$ ならば暗号文 $c'' \in \mathrm{TLWE}_{k_2}(\mu)$ を，
そうでなければ $c'' \in \mathrm{TLWE}_{k_2}(0)$ を返す．

begin
  $c' \leftarrow$ `ProgBootstrap`$(f_{\texttt{Bootstrap}}, c, \mathrm{BK})$
  $c'' \leftarrow (\mathbf{0}, \mu') + c'$
  return c''
end

アルゴリズム 4 も入力と出力で暗号文の鍵が異なるため，通常は [6] §4.1 の “Key Switching” を用いて入力と出力の鍵を同じものにする．Key Switching を行う場合については [6] Algorithm 10 の前後で詳しく述べられている．

4. 提案手法

本研究では秘密計算のモデルとしてクライアント・サーバモデルを考える．クライアント，サーバはともに 1 者の

みであり，全体で 2 者が参加する．提案方式では，クライアントはブートストラップ鍵を生成し，サーバへ送信しておく．その後，サーバは自身で生成した乱数とクライアントのブートストラップ鍵を用いて，ローカルでの計算のみによってベルヌーイ分布 $\mathrm{Ber}(p)$ に従う乱数を生成することができる．

4.1 ベルヌーイ分布に従う乱数の生成

提案プロトコルを Algorithm 5 に示す．このプロトコルのステップ 2–3 は準備で，ステップ 4–6 は本処理である．本処理を何回繰り返すとしても準備は一回行えば十分である．なお，ステップ 4 を行う代わりに $(\mathbb{Z}/2N\mathbb{Z})^{n+1}$ の元 $\bar{c}$ を一様ランダムに取ることでアルゴリズム `ProgBootstrap` で行う丸め処理を省略できるが，ここでは入力を揃えて Algorithm 5 のように書いた．

Algorithm 5: 提案プロトコル `EncryptedBer`

Input: 確率 $p \in \{\frac{0}{N}, \frac{1}{N}, \frac{2}{N}, \dots, \frac{N}{N}\}$，クライアントのブートストラップ鍵 $\mathrm{BK} = (\mathrm{BK}_1, \dots, \mathrm{BK}_n)$.

Output: サーバ: 確率 p で $\mathrm{msg}(c') = \frac{1}{2} \in \mathbb{T}$，確率 $1-p$ で $\mathrm{msg}(c') = 0 \in \mathbb{T}$ が成り立つ，クライアントの鍵で暗号化された TLWE 型暗号文 c'. クライアント: 無し.

1 **begin**
  // 準備
2 クライアント はブートストラップ鍵 $\mathrm{BK}_1, \dots, \mathrm{BK}_n$ を サーバ に送信する.
3 サーバ は写像 $f_{\texttt{EncBer}}$ を下記のものとする.
  // 本処理．以降は サーバ のみが計算を行う.
4 元 $c = (a_1, \dots, a_n, b) \in \mathbb{T}^{n+1}$ を一様ランダムに取る.
5 $c' \leftarrow \texttt{ProgBootstrap}(f_{\texttt{EncBer}}, c, \mathrm{BK})$.
6 **return** c'
7 **end**

提案プロトコル 5 (`EncryptedBer`) でアルゴリズム `ProgBootstrap` に入力する写像 $f_{\texttt{EncBer}} \colon \mathbb{Z}/2N\mathbb{Z} \to \mathbb{T}$ は次のものである．トーラス $\mathbb{T}$ では $-\frac{1}{2} \bmod \mathbb{Z} = \frac{1}{2} \bmod \mathbb{Z}$ が成り立つことに注意せよ．

$$f_{\texttt{EncBer}}(i) = \begin{cases} \frac{1}{2} \bmod \mathbb{Z} & (0 \le i \le pN-1) \\ -\frac{1}{2} \bmod \mathbb{Z} & (0+N \le i \le (pN-1)+N) \\ 0 & (\text{上記以外の場合}) \end{cases},$$

$p = \frac{0}{N}$ のときは $pN - 1 = -1$ となるため，$f_{\texttt{EncBer}}$ は定数関数 0 に等しくなることに注意せよ．この写像が $f(i+N) = -f(i)$ を満たすことは容易に確認できる．

ステップ 4 で生成したランダムな元 $c = (a_1, \dots, a_n, b) \in \mathbb{T}^{n+1}$ は，$\mathbb{T}$ の一様ランダムな元をクライアントの鍵で暗号化した TLWE 型暗号文と見ることができる．したがって $c \in \mathbb{T}^{n+1}$ をクライアントのブートストラップ鍵とともに通常の `Bootstrap` に渡せば，確率 $\frac{1}{2}$ で $\mathrm{msg}(c) = \frac{1}{2}$，確率 $\frac{1}{2}$ で $\mathrm{msg}(c) = 0$ が成り立つようなクライアントの鍵で暗号化された暗号文 c が得られる．

プロトコル `EncryptedBer` の正当性を示すために，まずアルゴリズム `ProgBootstrap` の正当性を証明する．[2] Theorem 3.1 でも示されているが，$f(i+N) = -f(i)$ の必要性しか示されていないため改めて証明する．

命題 2. 任意の $i \in \mathbb{Z}/2N\mathbb{Z}$ について $f(i-N) = -f(i)$ が成り立つ写像 $f \colon \mathbb{Z}/2N\mathbb{Z} \to \mathbb{T}$ を考える．$\mathbb{T}[X]$ の元 $v = \sum_{i=0}^{N-1} f(i) X^i$ と任意の $j \in \mathbb{Z}$ について，$\mathrm{Rep}(X^{-j} v) \in \mathbb{T}[X]$ の定数項は $f(j \bmod 2N\mathbb{Z})$ である．

（証明）． $j = j' + 2qN$ $(0 \le j' < 2N, q \in \mathbb{Z})$ と表すと，$\mathbb{T}_N[X]$ では $X^{2N} = 1$ なので $X^{-j} = X^{2(q+1)N} X^{-j} = X^{2N-j'}$ となる．$0 < 2N - j' \le 2N$ が成り立つことに注意せよ．

$0 \le j' \le N$ のとき，$X^{2N-j'} v$ を次数 N 以上・未満の項で分解すると以下のようになる．

$$X^N \left(\sum_{0 \le i < j'} f(i) X^{i-j'+N} \right) + \left(\sum_{j' \le i < N-1} f(i) X^{i-j'} \right)$$

このとき，$\mathrm{Rep}(X^{2N-j'} v) = -$(前の括弧内)+(後の括弧内) なので，$\mathrm{Rep}(X^{2N-j'} v)$ の定数項は $f(j' \bmod 2N\mathbb{Z})$ である．

同様に $N < j' < 2N$ のとき，$X^{2N-j'} v$ を次数 N 以上・未満の項で分解すると以下のようになる．

$$\left(\sum_{0 \le i < j'} f(i) X^{i-j'+2N} \right) + X^N \left(\sum_{j' \le i < N-1} f(i) X^{i-j'+N} \right)$$

このとき，$\mathrm{Rep}(X^{2N-j'} v) =$ (前の括弧内) $-$ (後の括弧内) なので，$\mathrm{Rep}(X^{2N-j'} v)$ の定数項は $-f(j'-N \bmod 2n\mathbb{Z}) = f(j' \bmod 2N\mathbb{Z})$ である．ここで $f(j' - N \bmod 2N\mathbb{Z}) = -f(j' \bmod 2N\mathbb{Z})$ を用いた． ∎

これを踏まえてプロトコル `EncryptedBer` の正当性を示す．

命題 3. プロトコル `EncryptedBer` の出力 c' について確率 p で $\mathrm{msg}(c') = \frac{1}{2} \in \mathbb{T}$，確率 $1-p$ で $\mathrm{msg}(c') = 0 \in \mathbb{T}$ が成り立つ．

（証明）． $\mathbb{Z}/2N\mathbb{Z}$ の全要素が一様な確率で $f_{\texttt{EncBer}}$ に渡される．

$p = \frac{0}{N}$ のとき，$f_{\texttt{EncBer}}$ は定数関数 0 に等しくなるため，プロトコル 5 の出力 c' は常に $\mathrm{msg}(c) = 0$ を満たす．

以下，$p > \frac{0}{N}$ とする．$\mathbb{Z}/2N\mathbb{Z}$ の要素のうち，$I = \{0, \dots, pN-1\} \cup \{0+N, \dots, (pN-1)+N\} \bmod 2N\mathbb{Z}$ に属す要素は，$f_{\texttt{EncBer}}$ で $\frac{1}{2} (= -\frac{1}{2}) \in \mathbb{T}$ に写される．よっ

てプロトコル 5 の出力 c' について $\mathrm{msg}(c') = \frac{1}{2} \in \mathbb{T}$ が成り立つ確率は

$$\frac{|I|}{|\mathbb{Z}/2N\mathbb{Z}|} = \frac{2pN}{2N} = p.$$

出力 c' について $\mathrm{msg}(c')$ が取りうる値は $0, \frac{1}{2} \in \mathbb{T}$ のどちらか一方である（排反事象である）から，$\mathrm{msg}(c') = 0 \in \mathbb{T}$ が成り立つ確率は $1 - p$. ∎

4.2 一様分布に従う乱数の生成

提案プロトコル `EncryptedBer` を用いれば，非対話で秘密の一様乱数を生成することも容易となる．ベルヌーイ分布 $\mathrm{Ber}(\frac{1}{2})$ に従う暗号化された乱数をプロトコル `EncryptedBer` で m 個生成し並べれば，区間 $[0, 2^m]$ 上の二進表現された一様乱数とみなすことができる．

5. 応用: 差分プライバシー達成のための非対話型二者間ノイズ生成

提案プロトコル `EncryptedBer` を用いて差分プライバシーを達成するためによく用いられる離散乱数を構成する方法を示す．この方法の安全性についてもシステムモデルと攻撃者モデルを定義して述べる．

5.1 システムと攻撃者のモデル

提案手法の安全性を考えるために，システムモデルと攻撃者モデルを定義し，可能な攻撃を考える．

5.1.1 システムモデル

本研究では秘密計算のモデルとしてクライアント・サーバモデルを考える．クライアント，サーバはともに 1 者のみであり，全体で 2 者が参加する．

クライアントはサーバに暗号化したデータを送信し，(ϵ, δ)-差分プライバシーを達成するためのノイズ付加を含む計算をサーバに委託する．この際，サーバはサーバの秘密情報（例えば機械学習モデルのパラメータや個人情報を含むデータベース）を用いて計算を行う．より具体的には，クライアントとサーバが行うやり取りは以下を想定する．

(1) クライアント C は TFHE の鍵を生成し，この鍵で秘密情報 X_C を暗号化する．
(2) C はブートストラップ鍵と暗号化した情報 $\mathtt{Enc}_C(X_C)$ をサーバ S に送信する．
(3) サーバ S は受信したデータ $\mathtt{Enc}_C(X_C)$ とサーバ自身の秘密情報 X_S を用いて統計計算などの処理を行う．処理結果を $\mathtt{Enc}_C(R)$ とする．
(4) C, S のデータに施された計算を S はすべて知っているため，S は (ϵ, δ)-差分プライバシーを達成するために用いるべきノイズの分布のパラメータを平文で計算できる．S は計算したパラメータを用いてノイズ N を生成する．
(5) 生成したノイズ N を (3) の処理結果 $\mathtt{Enc}_C(R)$ に加える．この結果を $\mathtt{Enc}_C(R + N)$ と表す．
(6) 処理結果 $\mathtt{Enc}_C(R + N)$ はサーバ S からクライアント C に送られる．
(7) C によって復号された処理結果 $R + N$ は，クライアントから直接伝えられるなどしてサーバ側も知ることができるとする．C が処理結果を不特定多数に公開することもあり得る．

5.1.2 攻撃者モデル

本研究では攻撃者のモデルとしてクライアントまたはサーバが持つ秘密情報を得ようとする Semi-honest な攻撃者を考える．

5.1.2.1 サーバへの攻撃

攻撃者はクライアントと結託して，(5) 以降の差分プライベートな処理結果 $R + N$ を入手できたとする．上記のシステムモデルでクライアントと結託した攻撃者が他に得られる情報はない．しかし $R + N$ は (ϵ, δ)-差分プライベートであり，(ϵ, δ)-差分プライベートが保証する範囲で安全性が保たれている．

5.1.2.2 クライアントへの攻撃

攻撃者はサーバと結託して次のすべてを入手できたとする．

(a) (2) でクライアントがサーバに送信した秘密情報 $\mathtt{Enc}_C(X_C)$,
(b) (3) でサーバが行った処理結果 $\mathtt{Enc}_C(R)$，及び処理途中の情報,
(c) (4) でサーバが生成したノイズ $\mathtt{Enc}_C(N)$，及びその部分情報,
(d) (5) 以降の (ϵ, δ)-差分プライバシーを満たす処理結果 $R + N$.

このうち (a), (b) はサーバ S が知らない鍵を使って TFHE で暗号化されているため，TFHE 暗号が安全ならば (a), (b) の情報を得ることは出来ない．また，攻撃者は (d) を平文で得ることができるが，(ϵ, δ)-差分プライベートが保証する範囲で安全性が保たれている．

しかしサーバと結託した攻撃者がノイズ N を平文で得られると，差分プライベートな処理結果 $R + N$ から差分プライベートでない処理結果 R が得られる．また攻撃者がノイズ N の部分情報を得た場合でも，攻撃者はノイズの部分情報と (ϵ, δ)-差分プライバシーを満たす処理結果 $R + N$ から，(ϵ, δ)-差分プライバシーを満たさない計算結果を攻撃者は得られる．そのため出力結果の安全性を保証するには，(4) で生成されたノイズ N 及びノイズの部分情報も保護する必要がある．

5.2 提案手法の応用

以下，差分プライバシーを達成するためによく用いられる離散乱数をベルヌーイ分布から構成する方法を示す．

通常，TFHE の TLWE 型暗号文では 0 または $\frac{1}{2}$ しか表

せない[*2]ため，TFHE で整数を表す際には必然的に二進表現を用いる．

5.2.1 二項分布に従う乱数の生成

二項分布 $\mathrm{Bin}(m,p)$ に従う乱数を生成するには，確率 p で 1 が出るベルヌーイ試行を m 回繰り返し，何個が $\frac{1}{2}$ の暗号文であるか数え上げれば良い．ベルヌーイ試行は提案プロトコル `EncryptedBer` で実現できる．

また，生成された暗号文のうち何個が $\frac{1}{2}$ の暗号文であるか数える問題は Hamming 重みあるいは population count の計算とみなすことができる．これはビット演算（シフトと AND）と $\lceil \log_2 m \rceil$ 回の整数加算で計算するアルゴリズムがよく知られている．

5.2.2 離散ラプラス分布に従う乱数の生成

離散ベルヌーイ分布 $\mathrm{DLap}(p)$ は台が $\mathbb{Z}$ であり，確率質量関数が $f_{\mathrm{DLap}(p)}(k) = \frac{1-p}{1+p}p^{|k|}$ で与えられる確率分布である．名前の通りラプラス分布の離散版として [11] で導入された．差分プライバシーを達成する離散値を出力する機構のなかで用いられる．

[17] で述べられているとおり，離散ベルヌーイ分布は $\mathrm{msg}(c_1)$ がベルヌーイ分布 $\mathrm{Ber}\left(\frac{1-p}{1+p}\right)$ に従い，$\mathrm{msg}(c_i)$ $(i>1)$ がベルヌーイ分布 $\mathrm{Ber}\left(\frac{1}{2}\right)$ に従う乱数列 $c_1, c_2, \ldots$ から生成できる．乱数列を作ったあとは，$\mathrm{msg}(c_i) = \frac{1}{2}$ である最小の i を計算する．これには例えば Wen らが提案した手法 LEAF [15] を用いることができるだろう．最後に $\frac{1}{2}$ の確率で i を，$\frac{1}{2}$ の確率で $-i$ を出力とすれば良い．

実際にはベルヌーイ分布に従う乱数を無数に生成することは出来ないため，生成できるのは離散ラプラス分布の両裾を切った分布に従う乱数である．このことが原因で，達成できる安全性は $\delta > 0$ についての (ϵ, δ)-差分プライバシーになる．δ を十分小さくするために必要なベルヌーイ分布に従う乱数の個数は江里口らの先行研究 [17] で評価されている．

5.3 提案手法の安全性

以上の 5.2 節で提案した手法では，プロトコル `EncryptedBer` で生成された乱数を入力とするアルゴリズムを実行することで乱数を生成する．プロトコル `EncryptedBer` で生成された乱数はサーバ S が知らないクライアントの鍵を用いて TFHE で暗号化されている上，アルゴリズムも暗号化した状態で行われている．したがって TFHE 暗号が安全である限り，5.1.2 節で定義した攻撃者は差分プライバシー達成のために使われたノイズについての情報を得ることが出来ない．

6. 既存手法との比較

様々な（つまり $p = \frac{1}{2}$ とは限らない）p についてベルヌーイ分布 $\mathrm{Ber}(p)$ に従う乱数を安全に生成する手法を比較する．この節で乱数といえばベルヌイー分布 $\mathrm{Ber}(p)$ に従うものとする．

Dwork らの先行研究 [8] ではベルヌーイ分布 $\mathrm{Ber}(1/2)$ に従う $2n\log(n+d)$ 個の乱数を入力すると，ベルヌーイ分布 $\mathrm{Ber}(p)$ に従う乱数を n 個生成する回路が提案されている．この [8] で提案されている回路の深さは $\Theta(\log(n+d))$，サイズは $\Theta(nds^2\log(n+d))$ となっている．ここで d は p の精度，s は回路のパラメータで小さな正整数である．したがって [8] の手法で乱数を n 個生成する場合に必要な通信量は $O(nds^2\log(n+d))$ と評価できる．

また，江里口らの先行研究 [17] では固定された集合上の一様分布に従う乱数と定数の大小比較を行って乱数を生成する．[17] では具体的には西田と大田 [12] のプロトコル Joint Random Number Sharing と Interval Test を用いて一様乱数の生成と大小比較を行っている．秘密計算は $\mathbb{Z}_p$（p は素数）上の加法秘密分散を用いて行われる．[12] のプロトコルを用いて乱数を n 個生成する場合，各パーティの通信量は $n(110\log_2 p + 1)\log_2 p$ ビット，ラウンド数は $13n$ となる．この手法の安全性は情報理論的安全性となっている．

一方，本稿での提案手法を用いて乱数を n 個生成する場合，[6] で述べられている TFHE のパラメータ（198 ビット安全性）で実装すると，通信量は 15.6MB，計算量は $\Theta(n)$ となる．提案手法は既存手法と異なり非対話型プロトコルなので，通信量は n に比例しないし，他の参加者の通信を待つ必要はない．

7. 結論

本稿では完全準同型暗号方式の一つ TFHE を用いた，ベルヌーイ分布 $\mathrm{Ber}(p)$ に従う乱数を安全に生成する方法を提案した．この手法は二者間プロトコルであり，事前準備のほかは通信を行わず，何個乱数を生成するとして事前準備は変わらない．また，この手法で生成された乱数について，参加者（特にサーバ）は事前に知り得る情報（パラメータ p の値）以上の情報は知り得ない．

提案方式の課題点は二つ挙げられる．一つは提案方式ではパラメータである確率 p の精度を上げることが容易ではないということである．確率 p の精度は TFHE のパラメータ N の大きさに比例するが，N を大きくするとブートストラップが遅くなる．また N としては 2 のべき乗しか取りえないため，N の値として現実的なものは $2^{10}, 2^{11}, 2^{12}$ ぐらいしかない．もう一つの課題点は，二項分布 $\mathrm{Bin}(m,p)$ に従う乱数を生成する際には $\lceil \log_2 m \rceil$ 回の整数加算が必要だが，TFHE ではこれが遅いということである．これ以上整数加算の回数を減らすアルゴリズムは知られていないため，高速に TFHE での整数加算を行う手法・実装の登場に期待するしかない．

[*2] 一つの TLWE 型暗号文で 1 ビットより多くの情報を表現する手法がいくつか提案されている [1], [3], [13].

参考文献

[1] Boura, C., Gama, N. and Georgieva, M.: Chimera: a unified framework for B/FV, TFHE and HEAAN fully homomorphic encryption and predictions for deep learning, Technical Report 758 (2018).

[2] Carpov, S., Izabachène, M. and Mollimard, V.: New techniques for Multi-value input Homomorphic Evaluation and Applications, Technical Report 622 (2018).

[3] Case, B. M., Gao, S., Hu, G. and Xu, Q.: Fully Homomorphic Encryption with k-bit Arithmetic Operations, Technical Report 521 (2019).

[4] Chillotti, I., Gama, N., Georgieva, M. and Izabachène, M.: Faster Fully Homomorphic Encryption: Bootstrapping in Less Than 0.1 Seconds, *Advances in Cryptology – ASIACRYPT 2016* (Cheon, J. H. and Takagi, T., eds.), Lecture Notes in Computer Science, Berlin, Heidelberg, Springer, pp. 3–33 (online), DOI: 10.1007/978-3-662-53887-6˙1 (2016).

[5] Chillotti, I., Gama, N., Georgieva, M. and Izabachène, M.: Faster Packed Homomorphic Operations and Efficient Circuit Bootstrapping for TFHE, *Advances in Cryptology – ASIACRYPT 2017* (Takagi, T. and Peyrin, T., eds.), Lecture Notes in Computer Science, Cham, Springer International Publishing, pp. 377–408 (online), DOI: 10.1007/978-3-319-70694-8˙14 (2017).

[6] Chillotti, I., Gama, N., Georgieva, M. and Izabachène, M.: TFHE: Fast Fully Homomorphic Encryption Over the Torus, *Journal of Cryptology*, Vol. 33, No. 1, pp. 34–91 (online), DOI: 10.1007/s00145-019-09319-x (2020).

[7] Dwork, C.: Differential Privacy, *Automata, Languages and Programming* (Bugliesi, M., Preneel, B., Sassone, V. and Wegener, I., eds.), Lecture Notes in Computer Science, Berlin, Heidelberg, Springer, pp. 1–12 (online), DOI: 10.1007/11787006˙1 (2006).

[8] 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).

[9] Froelicher, D., Troncoso-Pastoriza, J. R., Sousa, J. S. and Hubaux, J.: Drynx: Decentralized, Secure, Verifiable System for Statistical Queries and Machine Learning on Distributed Datasets, *IEEE Transactions on Information Forensics and Security*, Vol. 15, pp. 3035–3050 (online), DOI: 10.1109/TIFS.2020.2976612 (2020). tex.ids= FTSH20a conferenceName: IEEE Transactions on Information Forensics and Security.

[10] 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).

[11] 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).

[12] Nishide, T. and Ohta, K.: Multiparty Computation for Interval, Equality, and Comparison Without Bit-Decomposition Protocol, *Public Key Cryptography – PKC 2007* (Okamoto, T. and Wang, X., eds.), Lecture Notes in Computer Science, Berlin, Heidelberg, Springer, pp. 343–360 (online), DOI: 10.1007/978-3-540-71677-8˙23 (2007).

[13] Okada, H., Kiyomoto, S. and Cid, C.: Integerwise Functional Bootstrapping on TFHE, *Information Security* (Susilo, W., Deng, R. H., Guo, F., Li, Y. and Intan, R., eds.), Lecture Notes in Computer Science, Cham, Springer International Publishing, pp. 107–125 (online), DOI: 10.1007/978-3-030-62974-8˙7 (2020).

[14] Opanchuk, B.: NuFHE (2021). original-date: 2018-04-08T05:17:17Z.

[15] Wen, R., Yu, Y., Xie, X. and Zhang, Y.: LEAF: A Faster Secure Search Algorithm via Localization, Extraction, and Reconstruction, *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*, CCS '20, New York, NY, USA, Association for Computing Machinery, pp. 1219–1232 (online), DOI: 10.1145/3372297.3417237 (2020).

[16] 牛山 翔二郎，工藤 雅士，高橋 翼，井上 紘太朗，鈴木 拓也，山名早人：差分プライバシと準同型暗号の組合せに関する研究動向調査 (2020).

[17] 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，Kochi, Japan (2020).

正誤表

ページ	箇所	訂正前	訂正後
1	著者名	紀伊真昇，市川敦謙の二名のみ.	紀伊真昇，市川篤紀，千田浩司，濱田 浩気の四名. 所属はいずれも NTT セキュアプラットフォーム研究所.
1	所属	〒 180 – 0012	〒 180 – 8585
2	2.1 節	n 個の暗号文 $x_1, \ldots, x_n$	n 個の暗号文 $\mathtt{HE.Enc}_k(x_1), \ldots, \mathtt{HE.Enc}_k(x_n)$
2	2.2 節	「鋭敏度」	「敏感度」
3	3.1.1 節	平文 $m \in \{0, \frac{1}{2}\} = \frac{1}{2}$	平文 $m \in \{0, \frac{1}{2}\} = \frac{1}{2}\mathbb{B}$
4	3.2.1 節	s_i は $\mathrm{Rep}(S_i)$ の定数項	$s_{N(i-1)+j+1}$ は $\mathrm{Rep}(S_i) \in \mathbb{T}[X]$ での X^j の係数
4	Algorithm 2	$\mathrm{TLWE}_{k_0}(\frac{1}{2}b_0+)$	$\mathrm{TLWE}_{k_0}(\frac{1}{2}b_0 + \mathbb{Z})$
4	3.2.2 節	$f(m+\epsilon)$（ϵ は小さなノイズ） の暗号文を得ることができる.	$f(2Nm+\epsilon)$（ϵ は小さなノイズ，$2Nm+\epsilon$ は整数） の暗号文を得ることができる.
4	Algorithm 3	$f(\lceil 2Nm \rfloor + \epsilon)$	$f(2Nm+\epsilon)$
4	3.2.2 節	ノイズ ε 大きさ	ノイズ ε の大きさ