DEIM2020 E4-2(day1 p35)

TEEに基づく差分プライバシの検証

加藤　郁之† 曹　洋†† 吉川　正俊†††

† 京都大学大学院 情報学研究科　〒 606–8501 京都府京都市左京区吉田本町
††† 京都大学大学院情報学研究科　〒 606–8501 京都府京都市左京区吉田本町
††† 京都大学大学院情報学研究科　〒 606–8501 京都府京都市左京区吉田本町
E-mail: †fumiyuki@db.soc.i.kyoto-u.ac.jp, ††{yang,yoshikawa}@i.kyoto-u.ac.jp

あらまし　本研究では，局所差分プライバシにおけるプライバシ性の偽装の動機と偽装が行われることによる影響，信頼できない２者間におけるデータの受け渡しの中で，送信者から主張されるプライバシ性の検証を行うための方法についての提案を行う．TEE(Trusted Execution Environment) の実装の 1 つである Intel SGX を用いた方法によって，プライバシ保護のためにノイズを付加する摂動プロセスの完全性を検証する方法を提案する．実験では，SGX を用いてシステムの実装を行い定性的な評価を行うことで，本手法によって，信頼できない対象から得られたデータのプライバシ性の検証が可能であることを示す．

キーワード　Local Differential Privacy, Secure Hardware, Verifiable Computation

1 はじめに

差分プライバシ [12] は機密性の高い情報を大量に収集して，データのプライバシを保護しつつテーブル上の統計情報を公開するための厳密なプライバシの定義であり，標準的な技術になりつつある．従来の差分プライバシは主に静的なデータベースのレコードに対する集約関数を対象にして定義されている．出力に適切な大きさのノイズを付加することで，データベース内の秘匿レコードに対してプライバシを保護している．典型的な差分プライバシにおける信頼モデルは主に，個人情報を提供するクライアント（提供者），個人情報を収集するサーバ（収集者），サーバ内のレコードを元にして出力される統計情報を利用するユーザ（利用者），という 3 者間におけるモデルであり，信頼境界をサーバとユーザとの間に引いている．つまり，ユーザが信頼できない (malicious, untrusted) 対象として設定されているこれまでに差分プライバシを保証している統計的なアルゴリズムが数多く提案されている．[4] [13] [14] [23]．最近では，より信頼モデルを厳しく定義した局所差分プライバシ (Local Differential Privacy, LDP) [11] が提案されている．LDP においては，個人情報を収集するサーバも信頼できない対象として信頼モデルが設定されており，データを送信する前にクライアント側でランダム化してからサーバに送信している．また，より新しいモデルとして，サーバ側から信頼できない malicious なクライアントによるプライバシへの攻撃が注目されている．我々の研究では，この malicious なクライアントの振る舞いに注目する．

大量の個人から収集した個人情報を機械学習等に利用して知見を得ることは現在の IT サービスベンダにおいて一般的である．一方で，このことに対するプライバシの懸念は年々高まっている．個人情報の生データを信頼できないサーバに提供することに対するプライバシへの懸念から，LDP を用いたアプリケーションがいくつかの企業からすでにローンチされている．Google Chrome ブラウザに実装された PROPPER [13]，IOS の入力予測情報の収集 [1] などに対して、LDP に基づくプライバシ保護技術が導入されている．このような個人のデータ収集における LDP の導入は今後もより多くのサービスに広がっていくと見られる．また，情報市場や情報銀行等 [8] のモデルにおいても，LDP と同様に提供者側でプライバシ保護を行ってから情報を市場に提供するモデルが提案されている．また，プライバシと有用性のトレードオフを加味した個人情報のプライシングも議論されている [21]．

LDP は信頼できないサーバを仮定するモデルであるため，個人情報のプライバシ性はデータの提供者であるクライアント側がコントロールできることが望ましい．(以降，プライバシ性とはプライバシの強さの度合いを表すパラメータのこととする．) LDP はデータの所有者個々人がどの程度のプライバシを保護するかを決定してプライバシ性を設定できる技術である．ここが従来の差分プライバシと本質的に異なっており，注目すべき点である．我々はここで，個人情報の価値とプライバシ性のトレードオフ [19] [16] に基づき，クライアント側にこの差分プライバシのランダム性を利用したプライバシ性の偽装を行うモチベーションが発生すると考える．例えば，$1 - LDP$ と主張して実際には $3 - LDP$ のランダム化を行っている場合，よりプライバシ性は高くなるが，データの正確性は低下する．逆に言えば，嘘をついて低いプライバシ性を主張することでより大きな価値を求めることができる．malicious なクライアントのデータの偽装は、収集者のデータの分布の推定に対する攻撃にもなる [7] [6]．偽装されたプライバシ性によってデータが提供されることでサーバ側はデータの真の分布を正しく推定することが困難になり，データの有用性が落ちる．また、有用性が落ちるだけでなく、攻撃者がサーバ側の推定値をコントロールすることが可能になる。しかし，LDP では，収集された各データはランダム性アルゴリズムによってランダム化されているために，

受け取ったデータからプライバシ性の偽装を判定することはできない．

そこで本研究では，LDP を用いたプライベートなデータの収集において，malicious クライアントから収集するデータのプライバシ性の検証を目的とする．クライアントはランダム化されたデータと，そのデータが満たすプライバシ性をサーバ側に送信する．その下で，特定のプライバシ性を満たしていると主張されたランダム化されたデータに対して，そのデータを明らかにすることなくプライバシ性をサーバ側で検証可能である方法について議論する．つまり，プライベートなデータを送受信する 1 つの通信プロトコル内で秘匿性と検証性を満たすシステムについて考える．

本論文においてはハードウェアやプロセッサレベルでのセキュリティを完備する Secure Hardware [30] を用いた手法を考える．Trusted Execution Environment(TEE) [25] の実装の 1 つである Intel Software Guard Extensions(SGX) [20] [9] の Remote Attestation [15] を用いた方法を提案し，実験として提案システムの実装 [1]・評価を行う．具体的には，SGX に LDP のメカニズムの計算処理を委託することでプライバシ性の検証を可能にする．その摂動プロセスの完全性を Remote Attestation によってクライアント側から検証できるようにするというのが本提案手法の要旨である．評価においてはプライバシ性と検証可能性に加えて，TEE の設置対象についてサーバとクライアントの両者に対して検討を行う．加えて，今後の課題として，暗号・統計的手法などの，よりソフトウェア的な手法による課題の実現について考察を与える．

この論文における貢献を以下に示す：(1) 差分プライバシにおけるプライバシ性の検証の必要性を提起した (2) 上記の必要性について問題設定を行った (2) 上記の問題に対して Secrue Hardware を用いた解決策を提案し実装・評価を行った

本論文の構成は以下の通りである．2 節では，本研究の背景について述べる．3 節では，本研究に必要な事前知識について，4 節では，本研究における問題設定について 5 節では，提案手法とその実装・評価について述べる．6 節では，今後の課題とよりソフトウェア的な解決方法についての検討について，7 節では，まとめについて述べる．

2 研究背景

本節では，差分プライバシの検証必要性について述べる．

先に見た通り，差分プライバシの文脈において新たな信頼モデルが登場して malicious なクライアントの存在が議論の対象となっている．図 1 に，これまでのプライバシにおける典型的な信頼モデルの変遷を示す．

次に，malicious なクライアントの想定によって考えられるいくつかの問題について見る．

2.1 プライバシと有用性のトレードオフ

差分プライバシの概念が登場する以前から，データの匿名化

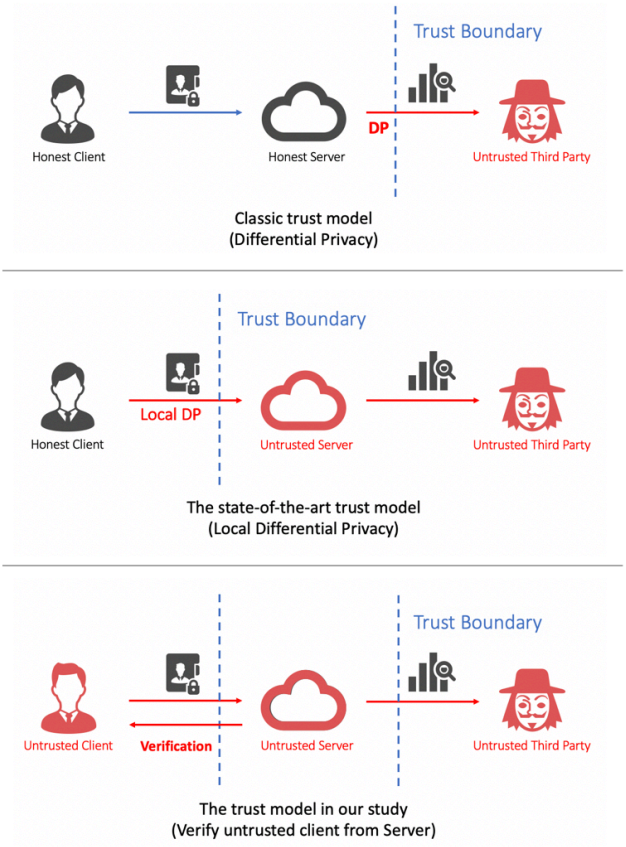

図 1　信頼モデルの変遷

に対してプライバシ性とデータの有用性のトレードオフに関する議論は行われている [19]．同様に，このトレードオフは差分プライバシにおいても考慮される必要がある [16]．この価値基準は，情報市場 [28] において重要な概念である．情報市場では，個人情報を市場に出す際にランダム化してプライバシを保護し，そのプライバシ性に応じて情報の価値が決定し価格づけされ，情報の提供者はその対価を得ることになる．その際のプライバシ性は個人情報の提供者が決定し，対価とプライバシーのトレードオフを自らが選択する事になる．このようなプライシングはアプリケーションに依存せずに様々な状況に見られ，プライバシと反比例するインセンティブ設計がなされることが多い．例えば，プライバシ性 ϵ に対して対価を $Reward = c \times \frac{1}{\sqrt{\epsilon}}$ とするような設計にすると，プライバシ性が低いほど対価を下げることになる．このように個人情報の提供者が自らプライバシ性を決定してデータを提供する場合，データの提供者はプライバシ性が低いと主張するほど高い対価・サービスを得られるようになる．つまり，クライアントに LDP のプライバシ性を偽装するインセンティブが働く．しかし LDP を満たすアルゴリズムは非決定的であり，収集したデータを見るだけではそのデータがクライアントが主張するメカニズムを通して本当に得られたものなのかを判断することは一般にできない．よって差分プライバシの検証が必要となると考えられる．このタイプのスキームでは，入力データの正しさは前提とする．入力データの正しさが保証されていないとこのようなスキームは成立しな

1：https://github.com/FumiyukiKato/verify-ldp

表 1　プライバシを偽装した際の Randomized Response の推定誤差

$\epsilon = 3$ に偽造した割合	分布の推定値
0	0.7500142
0.1	0.7374262
0.3	0.7136567
0.5	0.6892408

い．また，正しいデータを入力しないと適切なサービスを受けられないため，クライアントは正しいデータ入力が前提であることなども考えられる．

2.2　プライバシ性の偽装攻撃

LDP において，クライアントの一部が攻撃者となるもしくは攻撃者に乗っ取られて入力データを任意に改変されてしまう場合，Randomized Response や Heavey Hitter などの典型的ないくつかのアルゴリズムに対して脆弱性があることが Albert [7] らによって示された．攻撃者は入力データと対応するパラメータの偽装を行うことでサーバが計算したい推定値をコントロールすることができる．そのため，ここからも差分プライバシのプライバシ性の検証が必要と言える．

一定の割合のクライアントがプライバシを偽装して情報を提供した場合に有用性が落ちることは簡単な実験から確認することができる．$\epsilon - LDP$ を満たす LDP アルゴリズムの中で非常に簡単なものとして Randomized Response [31] がある．このメカニズムにおいては，以下のような $distortion\,matrix$ に従ってバイナリの出力をランダムに変えることで $\epsilon - LDP$ を保証する．

$$distortion\,matrix = \begin{pmatrix} \frac{e^\epsilon}{1+e^\epsilon} & \frac{1}{1+e^\epsilon} \\ \frac{1}{1+e^\epsilon} & \frac{e^\epsilon}{1+e^\epsilon} \end{pmatrix} \tag{1}$$

これに従い，もとの分布を下のように推定することができる．

$$\pi(\epsilon) = \frac{\lambda(e^\epsilon + 1) - 1}{e^\epsilon - 1} \tag{2}$$

$$\lambda : \text{observed distribution} \tag{3}$$

$$\pi : \text{estimated distribution} \tag{4}$$

設定としては，提供者は全員 $1 - LDP$ を満たしていると主張しており，一定の割合 (=r) の人間がプライバシ性を偽装して $3 - LDP$ を満たす Randomized Response を用いてランダム化を行っている．真の分布が $P(x = 1) = 0.75$ であり，収集対象者が 100 万である．結果は，表 1 のようになり，偽装者の割合が増えると推定値のズレも大きくなる．

2.3　そ　の　他

Federated Learning [17] は中央のサーバで学習モデルを保持しつつ，各クライアントのデバイスにそのモデルの一部をダウンロードする．各デバイスの中で個人データを用いて学習を行い，そのモデルのパラメータの差分を中央に送信することでプライベートに学習を行うスキームである．送信するパラメータの差分はクライアントのプライバシを保証する摂動プロセスを経てサーバに送信される．ここでは，malicious なクライアントに対してモデルの一部を提供することになるのでモデルに対して Poisoning Attack [3] 等の様々な攻撃が可能となる．また，malicious なクライアントは送信するパラメータの差分をコントロールすることで，上のパラメータ偽装と同じく中央のモデルを破壊することが可能である．ランダム化プロセスの検証に止まらずに malicious なクライアントの行動の検証必要性が求められる．

3　前 提 知 識

3.1　局所差分プライバシ

局所差分プライバシ (LDP) [11] は，従来の差分プライバシとは異なる．従来の差分プライバシは静的なレコードテーブルに対する分析や統計情報の公開に焦点を当てている．アウトプットをユーザに対して公開する際にランダム化メカニズムを適用している．しかし，LDP はデータの収集時に適用される．データの提供者はデータを送信する前に自らの環境でランダム化メカニズムをて適用し，ランダム化されたデータを収集者に対して送信する．従来の差分プライバシでは提供者は収集者を信頼しているため生データを渡すが，LDP では収集者を信頼していないため生データを直接渡すことはしない．これは，信頼境界をサーバとユーザの間に引くか，クライアントとサーバの間に引くかという信頼モデルの違いとして解釈することができる (図 1)．より厳しい信頼境界を引いていることが分かる．

LDP のモデルは以下のように定義される．クライアントが $x \in \mathcal{X}$ の秘匿データを持っているとする．$\mathcal{X}$ を秘匿データの定義域とし，$\mathcal{Y}$ を出力データの定義域とする．各クライアントは自分の環境にランダム化メカニズム $M : \mathcal{X} \to \mathcal{Y}$ を持っており，クライアントは事前に計算によって $y = M(x)$ を得る．クライアントはサーバに y を送信する．ここで，確率分布 $\mathbf{P}$ を用いて

定義 1. 任意の $(x, x') \in \mathcal{X}$, 任意の $Y \in \mathcal{Y}$ に対して

$$\mathbf{P}[M(x) \in Y] \leqq e^\epsilon \cdot \mathbf{P}[M(x') \in Y] + \delta$$

を満たす時，メカニズム M は (ϵ, δ)-local differential privacy を満たす．もし $\delta = 0$ ならば M は ϵ-local differential privacy(ϵ-ldp) を満たす．

一般に LDP モデルの定義は唯一ではなく，ユーザが複数いる場合や，より広い範囲のプロトコルで LDP を保証するような定義もありうる．モデルの定義の仕方は対象としているシナリオや問題設定によって異なる．本研究では，2 者間の 1 度の通信におけるランダム化メカニズムの検証を対象にしているため，上のような典型的な定義になっている．

3.2　Intel SGX

Intel Software Guard Extensions(Intel SGX, SGX) [20] [9] は Trusted Execution Environment(TEE) [25] の実装の 1 つである．TEE とは，計算機実行環境内に高いレベルのセキュリティで保護された環境を作成して，保護された環境内で情報の保護や秘匿計算，実行環境の検証などを行えるようにした機構

のことである．Intel SGX をはじめ，ARM の Trustzone [2] などいくつかのプロセッサベンダから TEE が実装されたプロセッサが商業的にローンチされている．また MIT Sanctum [10] は OSS として RISC-V の拡張機能として開発されている．Subramanyan [29] らは TEE の不可欠な特徴として，TEE 内のプログラムの改竄を検知することができる検証性，保護メモリ領域内のプログラムの実行結果が侵害されない完全性，意図しない情報が保護領域の外には一切漏れない機密性の 3 つの要素を提案している．また，Intel SGX も特定の条件を満たした攻撃者に対して，これらの性質を満たすとの証明を与えられた．

Intel SGX の実態は Intel の一部のプロセッサに搭載された特別な命令セットのことである．SGX の命令セットは特定の特権モードでしか実行することができないことがハードウェアレベルで保証されている．それらの特別な命令によって enclave と呼ばれる暗号化された保護領域をメモリ上に確保し，その保護領域上で計算を実行する．SGX のコア機能には OS やハイパーバイザ，root 権限を持った攻撃者なども干渉できないため，ソフトウェアレベルで enclave に対して攻撃を行うことはできないようになっている．ただし、SGX に対しては、サイドチャネル攻撃を中心に多くの脆弱性と攻撃手法が研究されており [18] [26] [5]、それらに対する改良についても研究が行われている [27] [24] [32].

3.2.1 Remote Attestation

TEE および SGX の重要な機能として Remote Attestation [15] がある．Remote Attestation とはリモート環境の enclave の実行環境および実行プログラムの動作の完全性を検証できる機能である．以下に概要を述べる。（図 2）検証者 (Service Provider, SP) は SGX を搭載したリモートの信頼できないアプリケーション (Independent Software Vendor) に対してリクエストを送信する。信頼できないアプリケーションはプラットフォーム内の enclave に対して SGX 実行環境のレポート作成を依頼する。検証者は受け取ったレポートを信頼できる検証機関 (Attestation Service) に送信し検証をリクエストする．この場合の検証機関はプロセッサベンダである Intel が提供する Intel Attestation Service(IAS)[2]である．IAS はプロセッサにハードコーディングされた秘密情報と対になっている非対称鍵を持っているため，enclave 内から IAS に対してセキュアにレポートを送信することが可能である．レポートの中身は，展開された enclave 環境の初期状態のハッシュ値，encalve をビルドした際にイメージにハードコーディングされたビルドユーザの署名などが格納されている．検証者は検証結果を受け取りリモートの enclave の動作の完全性を検証することができる．レポートは SGX 内の秘密鍵で暗号化されて署名されているので改竄や盗聴は暗号学的に不可能である．SGX 内の秘密鍵に対しては特定の Encalve イメージ内の命令から間接的にしかアクセスできないように制御されているため，不正に秘密鍵を取得することもできない．Remote Attestaion においては SP が enclave との鍵交換も同時に行っているので、以降検証者は

2 : https://software.intel.com/en-us/sgx/attestation-services

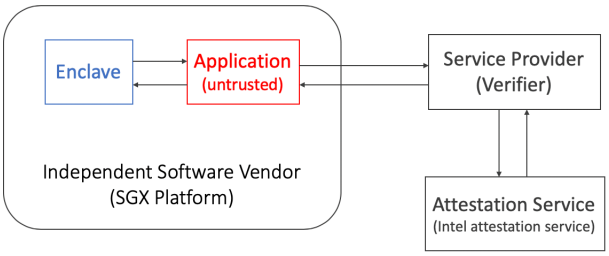

図 2　Remote Attestation

SGX に対してセキュアチャネルを張って任意の通信が可能となる．

4　問題設定

4.1　準　　備

本研究のシナリオでは，クライアントから中央のサーバが LDP を満たした個人情報データを収集する．その中でも，クライアントがサーバに対してデータを送信する 1 度の通信プロトコルを対象にしてプライバシ性の検証を可能にすることを目的とする．より一般的な設定として，n 人のクライアントに対してサーバがデータを収集して出力を出すというプロトコル Π で LDP を保証することを考えると以下のようになる: n 人のクライアント $i \in \{1, ..., n\}$ がそれぞれ個人情報 $x_i \in \mathcal{X}$ を持っている．クライアント i はそれぞれローカルの環境にメカニズム M_i を持っている．サーバ S はクライアントの出力データ $(y_1, ..., y_n)$ を受け取り，値を出力する関数を持つ．つまり $S : \mathcal{Y}^n \to \mathcal{Z}$ で $\mathcal{Z}$ は S の出力がとりうる値の集合である．プロトコルは $\Pi = ((M_1, ..., M_n), S)$ というタプルで定義される．このとき，以下のようにプロトコル全体の LDP を定義できる．

定義 2. ([7], Definition 2.1) 任意の $i \in \{1, ..., n\}$，任意の $(x, x') \in \mathcal{X}$，任意の $Y \in \mathcal{Y}$ に対して

$$\mathop{\mathbf{P}}_{M_i}[M_i(x) \in Y] \leqq e^{\epsilon} \cdot \mathop{\mathbf{P}}_{M_i}[M_i(x') \in Y] + \delta$$

を満たす時，プロトコル $\Pi = ((M_1, ..., M_n), S)$ は (ϵ, δ)-local differential privacy を満たす．もし $\delta = 0$ ならば Π は ϵ-local differential privacy を満たす．

また，これと定義 1 を合わせて以下の定理が自明に導かれる．

定理 1. 全てのクライアントのメカニズム M に対して (ϵ, δ)-*local differential privacy* を満たすならばプロトコル全体で (ϵ, δ)-*local differential privacy* を満たす

よってクライアントとサーバ 2 者間の 1 回の通信のメカニズムのプライバシ性の検証が可能であればプロトコル全体の検証も可能である．したがって本研究の問題設定としてはクライアントサーバ間の 1 度の通信プロトコルを対象とする．

4.2　問題設定

機密データの提供者であるクライアントを C，その機密デー

タを $\boldsymbol{x} \in \mathcal{X}$，摂動を行うメカニズムを $\mathcal{M}$，メカニズム適用後のランダム化された秘匿データを $\boldsymbol{y} \in \mathcal{Y}$，データの収集者であるサーバを S とする．ここで，$\mathcal{M}$ はプライバシバジェット $\epsilon > 0$ とデータを引数に取ってランダム化したデータを返す関数であり，$\mathcal{M} : \mathbb{R} \times \mathcal{X} \to \mathcal{Y}$ である．また，メカニズム $\mathcal{M}$ は C と S の両者がその詳細を知っている．$\mathcal{M}$ は，第一引数に ϵ が与えられたとき，任意の $\epsilon > 0$ に対し，ϵ-*ldp* を満たすメカニズムとなる．そして第 2 引数に与えられたデータをランダム化してデータ ($\in \mathcal{Y}$) を出力する．局所差分プライバシでは，C は S に対して生データ $\boldsymbol{x}$ を公開せずに，$\boldsymbol{y}$ と ϵ のみを公開する．その上で，S は C のランダム化後のデータ $\boldsymbol{y}$ が本当に $\mathcal{M}(\epsilon, \boldsymbol{x}) = \boldsymbol{y}$ であるかということを検証する．つまり，S が受け取ったデータが，C が主張するメカニズムのみが実行された状態であるかを確かめる．これらは，クライアントがデータを入力してサーバがランダム化されたデータを得るまでを一連のプロトコルとして以下のタプルで与えられる．

$$\Psi = (C, \boldsymbol{x}, S, \boldsymbol{y}, \mathcal{M}, \epsilon) \tag{5}$$

よってまとめると，プロトコル Ψ は以下の 2 つの性質を満たす必要がある．

検証可能性 S は公開情報 $\boldsymbol{y}$ と ϵ からメカニズム $\mathcal{M}$ が正しいパラメータ ϵ を与えられてクライアントの入力に対して実行された結果が $\boldsymbol{y}$ であることを十分に高い確率 p_{high} で確信できる．すなわち以下のような条件を満たす．ただし，S は次の機密性の条件を満たすために $\boldsymbol{x}$ についての知識を持つことはできない．

$$\mathbf{P}[\mathcal{M}(\epsilon, \boldsymbol{x}) = \boldsymbol{y}] \geqq p_{high} \tag{6}$$

機密性 S は公開情報 $\boldsymbol{y}$ と ϵ 以外の機密データ $\boldsymbol{x}$ に関する情報を得られない．つまり，以下のように，S がプロトコル Ψ によって観測できる情報 $obs(\Psi)$ 得ても，ϵ-*ldp* が満たす

$$\forall \boldsymbol{x}, \boldsymbol{x'} \in \mathcal{X}$$
$$\mathbf{P}[\mathcal{M}(\boldsymbol{x}) = \boldsymbol{y} | obs(\Psi)] \leqq e^{\epsilon} \cdot \mathbf{P}[\mathcal{M}(\boldsymbol{x'}) = \boldsymbol{y} | obs(\Psi)] \tag{7}$$

本論文における問題設定は，上に述べた問題設定を簡単化して以下のように制限を加える．データはバイナリとして，メカニズムは Randomized Response に固定する．

$$\mathcal{X}, \mathcal{Y} = \{0, 1\} \tag{8}$$
$$\mathcal{M}\,(\mathbb{R} \times \mathcal{X} \to \mathcal{Y}) : \text{Randomized Response} \tag{9}$$

これらの設定の下で，プロトコル Ψ において，検証可能性 (式 6) と機密性 (式 7) を満たすことを目的とする．

これらは通常は相反する性質であり，極端な例を考えると，なんの工夫もない通常の LDP であれば秘匿性は満たされるが，当然検証可能性を満たすことはできない．生データをサーバに送ってメカニズムを単純にサーバ側で実行するようにすれば，検証可能性を満たすが，秘匿性を満たすことはできない．しかし，これらを同時に実現することが必要である．

5 提 案 手 法

3 節で述べた Intel SGX の Remote Attestation [15] を用いた方法を提案し，実験として実装・評価を行う．LDP のメカニズムの計算処理を SGX に委託して SGX 内で処理を行うことでこれを実現するが，実現の仕方には 2 つのバリエーションがあると考える．サーバ側に SGX を置いたシステムとクライアント側に SGX を置いたシステムである．これらをそれぞれ S_s と S_c とする．

S_s では，中央のサーバのみに SGX が必要である．そのため，実利的には導入しやすい型と言える．クライアント側に encalve の実行イメージを送る必要も無い．簡略化した実行フローは以下のようになる．

1. クライアントからプロトコルの開始のリクエストが来て，サーバはクライアントに対しては encalve の完全性を示すレポートを発行して送信する．
2. クライアントはサーバから送られてくる暗号化されたレポートを IAS に対して検証依頼を行い，Remote Attesation によってサーバサイドの enclave 内の検証を行う．
3. enclave 内の検証が終わると同時に，サーバ上の SGX と ECDHE によって鍵交換が完了する．
4. 秘密情報から共通鍵を作成してクライアントサイドのデータを暗号化して，プライバシバジェットと共にリモートの SGX 環境に送る．
5. 暗号化されたデータは SGX 内で復号される．また SGX 内で，与えられたプライバシバジェットを使用してメカニズムを適用する．メカニズムでは SGX の真正乱数生成モジュールを用いて発生させた乱数によってランダム化される．
6. クライアントから送信されたプライバシバジェットとランダム化されたデータは，信頼できないメモリ領域にコピーされてサーバに取り出される．

同様に，S_c では各クライアントに SGX が必要である．サーバでビルドした encalve イメージを予めクライアント側に配布する必要がある．

1. クライアントはプロトコル開始のリクエストを送信して，サーバはクライアントに対しては encalve の完全性を示すレポートを依頼する．
2. クライアントはサーバに対してクライアントの encalve 環境の完全性を証明する暗号化されたレポートを送信する．サーバは IAS に対して検証依頼を行い，Remote Attesation によってクライアント側の enclave 内の検証を行う．
3. enclave 内の検証が終わると同時に，SGX と鍵交換が完了する．
4. クライアントは enclave に対して機密データとプライバシバジェットを入力する
5. 入力されたプライバシバジェットを用いて encalve 内でメカニズムを適用してデータをランダム化する．
6. プライバシバジェットとランダム化したデータは暗号化されてサーバに送信される．
7. サーバはクライアントから受け取ったデータを復号してランダム化したデータとプライバシバジェットを得る．

解決したい信頼モデルの話を振り返ると，S_c で実装を行う方が直感的である．信頼できないクライアントの動作をサーバ側が検証するというシナリオであるので，TEE をクライアント側におくことで，サーバは信頼していないクライアント側に信

頼できる領域を広げている．S_s では，信頼できないクライアントの動作をサーバ側の enclave 内で行うことを強制しているという解釈ができる．その代わりに，クライアントに enclave の検証を可能にしている．ただし，S_s は導入する TEE がサーバ側だけで済むため，比較的実装が容易である．また，S_s の方式は，計算をリモートの TEE にプライベートに委託するという TEE を用いた一般的な秘匿計算のスキームに近い．

5.1 実　　装

Intel Xeon E-2174G プロセッサを用いて Ubuntu16.04 の OS 上に Intel 社の提供する sgxsdk を用いて本システムの実装を行った (https://github.com/FumiyukiKato/verify-ldp)．実装したシステムは，サーバがリクエストを受け付けており，クライアントから TCP 上で SSL セッションを貼り，SSL 上で上のプロトコルにしたがってランダム化データを通信する．クライアントは，プライバシバジェットとデータを入力し，サーバはプライバシバジェットとランダム化データを得る．enclave を検証する側はプロトコルごとに実際に IAS と通信して Remote Attestation を行い，レポートを確認して検証を行う．

5.2 評　　価

本システムが満たすべき 2 つの条件，検証可能性と秘匿性について見る．

検証可能性

検証可能性は以下のようになる．サーバ・クライアントに関わらず，検証を行う側はリモートの enclave に対して Remote Attestation を行って enclave の初期状態を検証する．enclave の初期状態はメモリ状態などから計算されたハッシュ値によって表現される．このハッシュ値を H とする．SGX では MRENCLAVE という構造体がこれに相当する．H は IAS 以外には復号できない形で，SGX の特殊な命令で暗号化されている．Remote Attestation が成功して IAS 側から復号された H' が返ってくる．検証者はこの IAS から返ってきた H' を確認することができる．H および H' は enclave イメージのビルド時に決定されるマシンに非依存の値である．よって enclave イメージのビルド後にイメージの書き換えなどを行うと H' が変更し改竄が検知される．また起動時に enclave 内のメモリ状態の書き換えなどを行っても H' が変更されて改竄が検知される．よってこの H' が改竄されていなければ暗号学的に安全な強度で期待通りの動作が行われる．ここで，H' の変更が無いことの確認が行えたとすると，enclave 内にプライバシ性のパラメータとデータが渡されている，それらに対して enclave 内にプログラムされたメカニズムが正しく適用されている，サーバ側がランダム化された値を手にする，この一連の動作が暗号学的に安全な確率で保証されるので，検証可能性 (式 6) は満たされる．しかし，H' の確認するためには事前に正しい H' を手に入れておく必要がある．S_s と S_c を比較した場合，S_c については，サーバ側がクライアント側で動かす enclave イメージをビルドして，事前に正しい H' を取得してからクライアントにイメージを配布することで Remote Attestation によってクライアント上で実行される enclave の H' の確認を行うことができる．すなわち，検証可能性が満たされる．対して S_s の場合は，クライアントは正しい H' を事前に得ることはできない．S_s では，encalve 内のプログラムがサーバ上に存在しており，その動作がクライアントに透過的でないため，サーバの動作を透過的に検証することができない．透過的にというのは，プログラムの動作が確認できるという意味である．つまり，サーバの enclave の正常な動作の検証はできてもサーバが主張する動作が本当に行われているかどうかは検証できない，という状況が発生する．例えば，サーバ上の enclave 内で Randomized Response を用いてデータをプライベートに収集するサービスをローンチしたと考える．このとき，サーバは Remote Attestation による検証のために H' をクライアントに対して公開する．加えてサーバは，その H' は Randomized Response のみが正しく動作しているシステムであると主張する．クライアントは Remote Attestation を用いて H' を検証することで検証可能性を満たすことができる．この例だと，実行時やローンチ後の不正の有無はクライアント側で検証可能であるが，最初からサーバ側が不正を働いていた場合はクライアント側は検知できない．透過性がサーバ側の主張のみに依存している点が問題である．よって，S_s と S_c を比較すると，両者ともに enclave の起動後の改竄や攻撃に対しては検証可能であるが，S_s の場合は，透過的な検証可能性がなく，ビルド時点におけるサーバ側の不正に対して検証可能ではない．S_c の場合は，起動後の改竄や攻撃に対してもビルド時のプログラムの透過性についても検証可能である．

機密性

機密性については, [7] より，enclave 内におけるプログラムの実行からはキャッシュ攻撃を含むサイドチャネル攻撃が可能でない攻撃者に対しては，開発者が意図したメモリ領域以外の情報を観測できない．正しくプログラムを書いた場合は，$obs(\Psi) = \{\boldsymbol{y}, \epsilon\}$ でしかない．よって，プロトコル内で機密性 (式 7) は満たされる．

5.3 攻撃の分析

ここでは，本手法を用いてプライバシ性の検証を行うことでどのような攻撃に対して有効な防御となるのかをまとめる．

攻撃者のモデルとしては，TEE を用いているので，データとプライバシ性を任意に入力できる能力のみを有するとすることができる．メカニズムが確実に適用されるので攻撃者は任意の指定したデータをサーバに送信することはできない．

攻撃の 1 つは，情報市場に類似するスキームおいて LDP におけるプライバシ性を偽装することでデータをより大きな価値に偽装することである．この攻撃のモデルでは，入力は正しいデータに限る．スキーム自体で入力が正しいデータであることを前提としている．また，偽の入力データが入力されることは，データの提供者のプライバシを上げることには貢献するが，データの価値を大きくすることにつながらない．本手法ではデータとパラメータが入力された時点で，そのパラメータを用いてメカニズムが適用されてランダム化を行い，使用されたランダム化のパラメータが確実にサーバ側に送られる．よって

この攻撃は防ぐことができる．

もう 1 つの攻撃は，LDP で収集したデータに対するサーバ側の推定値をコントロールする攻撃である．[7] で示された通り，推定値は任意の入力が与えられることで攻撃者にコントロールされる可能性がある．先の攻撃とは異なり，入力されるデータとプライバシ性のどちらも考慮する必要がある．この攻撃の場合は，入力されるデータが正しいデータに強制されている場合は，プライバシ性を検証することで推定値を確実なものにすることが可能である．しかし，攻撃者が偽のデータを入力するシチュエーションだと，プライバシ性の検証だけではこの攻撃を防ぐことはできない．たとえメカニズムの適用を確実にしても本来の分布には影響しないからである．この攻撃を防ぐためには，正しい入力データの強制を行う仕組みが別に必要である．逆に正しい入力データが強制される状況で，プライバシ性が偽装されてしまう場合を考えると，本当のデータと不当に大きくずれたデータが高い信頼性を持ってサーバに収集されることになる推定値が壊れてしまう．よって，この攻撃については，任意の条件に対処するためにはプライバシ性の検証に加えて入力データの検証が必要である．

5.4 入力データの検証

先に述べた通り，入力データの検証はプライバシ性の検証がより幅広い問題に適用されるために必要である．一般に入力データの正しさは検証が困難であり，ゲーム理論によって正しい情報を入力させる方法などが研究されている [22]．入力データの中でも客観的なデータに対しては事実を Ground Truth として検証を行える可能性がある．我々は本研究に補足的に入力データの検証を行うスキームを考える．アイデアとしては，TEE の内部で入力の生データを暗号化してサーバに保存しておく．本システムの Remote Attestation 後に以下のフローを差し込む．

1. クライアントは TEE にデータとプライバシバジェットに加えて鍵 K_c を入力する．
2. TEE の中で K_c を用いて生データを暗号化してサーバに送信する．
3. サーバは暗号化された生データを保存しておく．

サーバは暗号化した生データを保存しておくことでクライアントに鍵 K_c を要求して入力データを復号して確認することができる．一方でサーバに保存されるデータは暗号化されているのでクライアントのプライバシは守られている．これによって防げることは，入力データの偽装の証拠が残ることになるので malicious と疑われるクライアントが発見できれば確実に偽装を検証することができる．これが malicious なクライアントの偽装行動の抑制になる．

5.5 実 利 性

現状の実利的な課題としては，SGX で Remote Attestation を用いると，クライアント・サーバ間の通信，IAS との通信が合計 10 回程度必要であり，任意のやり取りにやや時間がかかってしまう．また，全体のシステムとして IAS が単一のボトルネックになってスケールしない可能性がある．

6 今後の課題

6.1 手法の拡張

本論文においては，プロトコルは非常に単純化され，入出力の定義域はバイナリに対してのみでメカニズムは Randmized Response を固定していた．これをより一般的なアルゴリズムやスケーラビリティの必要なデータなどに対して適用すると実装面や検証性に問題が出てくる可能性がある．より高度な応用プロトコルに対するプライバシ性の検証への本手法の適用が今後の課題の 1 つである．また，プライバシ性の検証に限らず，malicious なクライアントの行動検証には，TEE を使用することがプラクティカルな解決策の 1 つになる．手法のみに注目すれば，2.3 で課題にあげた Federated Learning における malicious クライアントの行動検証などに応用することもできる．

6.2 その他の手法

本論文における TEE を用いた手法は，TEE を実装したハードウェアが必須であり，かつ，IAS という信頼できる第三者を仮定する必要がある．特に，後者の IAS を無条件に信頼しなくてはならないという条件は厳しい．我々はこの手法をハードウェア的な解決策と考えている．これらをソフトウェア的な手法に置き換えることが今後の課題の 1 つである．ソフトウェア的な解決策として統計的手法やゼロ知識証明が考えられる．本研究のシナリオでは，一回の通信プロトコルにおけるプライバシ性の検証を対象にしているため，統計的手法とは相性が悪いと考えている．統計的手法を用いるならば，出力のサンプリングを何度も行うことで分布の推定を行い，どの程度のランダム性が認められるのかを検証する方法が考えられる．しかし，1 クライアントに対して何度もデータを収集すると結局プライバシバジェットを多く消費してしまうことになるのでこのような手法は現実的ではない．ゼロ知識証明による解決策は本手法における厳しい仮定を取り除くことができる可能性があると考えている．ランダム性をゼロ知識証明によって検証することが目的となる．ゼロ知識証明は一般に 2 者間で完結するプロトコルであるので TTP の仮定も必要なく，TEE も使用せずにソフトウェアのみで実装できる点が優れている．ただし一般にゼロ知識証明は計算コストを伴うので，TEE による解決策と比較すると長所と短所があると考えている．よっていくつかのアルゴリズムに対して TEE による手法とゼロ知識証明による手法を比較することが必要である．

7 ま と め

本研究では，局所差分プライバシにおけるプライバシ性の偽装の動機と偽装が行われることによる影響，信頼できない 2 者間におけるデータの受け渡しの中で，送信者から主張されるプライバシ性の検証を行うための方法についての提案した．Remote Attestaion を用いた方法によって，プライバシ保護の

ためにノイズを付加する摂動プロセスの完全性を検証する方法を提案した．実験では，SGX を用いてシステムの実装を行い定性的な評価を行うことで，本手法によって，信頼できない対象から得られたデータのプライバシ性の検証が可能であることを示した．今後は，上にあげた課題について検討することを考えている．

文　献

[1] Learning new words. Apple Inc., Cupertino, CA (US) .March 2017. US Patent 9,594,741 B1.

[2] T. Alves. Trustzone: Integrated hardware and software security. *White paper*, 2004.

[3] E. Bagdasaryan, A. Veit, Y. Hua, D. Estrin, and V. Shmatikov. How to backdoor federated learning. *arXiv preprint arXiv:1807.00459*, 2018.

[4] R. Bassily and A. Smith. Local, private, efficient protocols for succinct histograms. In *Proceedings of the forty-seventh annual ACM symposium on Theory of computing*, pages 127–135. ACM, 2015.

[5] F. Brasser, U. Müller, A. Dmitrienko, K. Kostiainen, S. Capkun, and A.-R. Sadeghi. Software grand exposure:{SGX} cache attacks are practical. In *11th {USENIX} Workshop on Offensive Technologies ({WOOT} 17)*, 2017.

[6] X. Cao, J. Jia, and N. Z. Gong. Data poisoning attacks to local differential privacy protocols. *arXiv preprint arXiv:1911.02046*, 2019.

[7] A. Cheu, A. Smith, and J. Ullman. Manipulation attacks in local differential privacy. *arXiv preprint arXiv:1909.09630*, 2019.

[8] J. P. Choi, D.-S. Jeon, and B.-C. Kim. Privacy and personal data collection with information externalities. *Journal of Public Economics*, 173:113–124, 2019.

[9] V. Costan and S. Devadas. Intel sgx explained. *IACR Cryptology ePrint Archive*, 2016(086):1–118, 2016.

[10] V. Costan, I. Lebedev, and S. Devadas. Sanctum: Minimal hardware extensions for strong software isolation. In *25th {USENIX} Security Symposium ({USENIX} Security 16)*, pages 857–874, 2016.

[11] J. C. Duchi, M. I. Jordan, and M. J. Wainwright. Local privacy and statistical minimax rates. In *2013 IEEE 54th Annual Symposium on Foundations of Computer Science*, pages 429–438. IEEE, 2013.

[12] C. Dwork, F. McSherry, K. Nissim, and A. Smith. Calibrating noise to sensitivity in private data analysis. In *Theory of cryptography conference*, pages 265–284. Springer, 2006.

[13] Ú. Erlingsson, V. Pihur, and A. Korolova. Rappor: Randomized aggregatable privacy-preserving ordinal response. In *Proceedings of the 2014 ACM SIGSAC conference on computer and communications security*, pages 1054–1067. ACM, 2014.

[14] G. Fanti, V. Pihur, and Ú. Erlingsson. Building a rappor with the unknown: Privacy-preserving learning of associations and data dictionaries. *Proceedings on Privacy Enhancing Technologies*, 2016(3):41–61, 2016.

[15] A. Francillon, Q. Nguyen, K. B. Rasmussen, and G. Tsudik. A minimalist approach to remote attestation. In *2014 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pages 1–6. IEEE, 2014.

[16] X. He, A. Machanavajjhala, and B. Ding. Blowfish privacy: Tuning privacy-utility trade-offs using policies. In *Proceedings of the 2014 ACM SIGMOD international conference on Management of data*, pages 1447–1458. ACM, 2014.

[17] P. Kairouz, H. B. McMahan, B. Avent, A. Bellet, M. Bennis, A. N. Bhagoji, K. Bonawitz, Z. Charles, G. Cormode, R. Cummings, et al. Advances and open problems in federated learning. *arXiv preprint arXiv:1912.04977*, 2019.

[18] S. Lee, M.-W. Shih, P. Gera, T. Kim, H. Kim, and M. Peinado. Inferring fine-grained control flow inside {SGX} enclaves with branch shadowing. In *26th {USENIX} Security Symposium ({USENIX} Security 17)*, pages 557–574, 2017.

[19] T. Li and N. Li. On the tradeoff between privacy and utility in data publishing. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 517–526. ACM, 2009.

[20] F. McKeen, I. Alexandrovich, A. Berenzon, C. V. Rozas, H. Shafi, V. Shanbhogue, and U. R. Savagaonkar. Innovative instructions and software model for isolated execution. *Hasp@ isca*, 10(1), 2013.

[21] R. Nget, Y. Cao, and M. Yoshikawa. How to balance privacy and money through pricing mechanism in personal data market. *Proceedings of the SIGIR 2017 eCom workshop*, 2017.

[22] D. Prelec. A bayesian truth serum for subjective data. *science*, 306(5695):462–466, 2004.

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

[24] A. Rane, C. Lin, and M. Tiwari. Raccoon: Closing digital side-channels through obfuscated execution. In *24th {USENIX} Security Symposium ({USENIX} Security 15)*, pages 431–446, 2015.

[25] M. Sabt, M. Achemlal, and A. Bouabdallah. Trusted execution environment: what it is, and what it is not. In *2015 IEEE Trustcom/BigDataSE/ISPA*, volume 1, pages 57–64. IEEE, 2015.

[26] M. Schwarz, S. Weiser, D. Gruss, C. Maurice, and S. Mangard. Malware guard extension: Using sgx to conceal cache attacks. In *International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment*, pages 3–24. Springer, 2017.

[27] M.-W. Shih, S. Lee, T. Kim, and M. Peinado. T-sgx: Eradicating controlled-channel attacks against enclave programs. In *NDSS*, 2017.

[28] S. Spiekermann, A. Acquisti, R. Böhme, and K.-L. Hui. The challenges of personal data markets and privacy. *Electronic markets*, 25(2):161–167, 2015.

[29] P. Subramanyan, R. Sinha, I. Lebedev, S. Devadas, and S. A. Seshia. A formal foundation for secure remote execution of enclaves. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pages 2435–2450, 2017.

[30] M. Tehranipoor and C. Wang. *Introduction to hardware security and trust*. Springer Science & Business Media, 2011.

[31] S. L. Warner. Randomized response: A survey technique for eliminating evasive answer bias. *Journal of the American Statistical Association*, 60(309):63–69, 1965.

[32] S. Weiser and M. Werner. Sgxio: Generic trusted i/o path for intel sgx. In *Proceedings of the Seventh ACM on Conference on Data and Application Security and Privacy*, pages 261–268, 2017.