Vol.2014-GN-92 No.4
Vol.2014-SPT-9 No.4
2014/5/15

プライバシを考慮したクラウド型 IME

川本 淳平[1,a)]　櫻井 幸一[1,b)]

概要：クラウドサービスは便利である一方プライバシの問題は未だ重要な懸念事項の一つである．本論文では，昨今話題となったクラウド型 IME に焦点を当て，プライバシを保護したクラウド型 IME を提案する．IME は，入力記号列を受け取り別の記号列へ変換するソフトウェアと考えることができる．これをクラウドサービスとして実現する場合，利用者は入力記号列をクエリとしてクラウド上のサーバへ送信し，変換後の記号列を受け取るシステムと言える．本論文では，このモデルに適した検索可能暗号を提案し，またプライバシを考慮したクラウド型 IME への応用を示す．

1. はじめに

クラウドサービスの利点は数多くあり積極的に導入が進んでいるが，プライバシの問題は未だクラウドサービスにおける重要問題の一つである．本論文では，クラウドサービスの中でもクラウド型 IME (Input Method Editor) に着目し，利用者のプライバシを保護しつつクラウド型 IME サービスを利用する仕組みを提案する．本論文でいう IME とは，利用者が日本語や中国語，韓国語などを入力することを補助するアプリケーションである．ここでは，入力としてあるアルファベットからなる文字列を受け取り，対象となる変換後の文字列を返すシステムとして定義する．例えば，日本語の場合「konnichiwa」を受け取り「こんにちは」を返すシステムである．この IME をクラウドアプリケーションとして実現することを考える．

クラウド型 IME では，利用者は入力文字列をクラウドサーバへ送信する．入力を受け取ったサーバは，対応する変換後の文字列を計算し利用者へと送信する．クラウド型アプリケーションとして IME を実現すると，入力と変換後の文字列の対応表がクラウドサーバ上で管理されるため対応表の更新が容易になり，新語への対応が迅速になる．また，利用者が入力する文字列を収集することで，流行語の発見なども期待できる．

しかしながら，クラウド型 IME には大きなプライバシの問題がある．すなわち，利用者が入力する文字列は時として機微情報であるため，クラウドサーバへそれらを送信することは利用者が秘密にしていたい情報を公開することにつながる恐れがある．実際，あるクラウド型 IME において，利用者がクラウドサーバへの情報送信をオフにしていたにも関わらず，無断でサーバへ送信していた事例が問題となったことは記憶に新しい[*1]．我々は，利用者のプライバシに考慮したクラウド型 IME を実現するためには，クラウドサーバへは利用者の入力文字列を秘匿しつつ，アプリケーションの動作を実現する仕組みが必要であると考える．

本論文では，利用者のプライバシを考慮したクラウド型 IME の一モデルを提案し，範囲検索可能暗号を用いた実現方法を述べる．我々の IME は単純な入力文字列と変換後の文字列の対応表ではなく，利用者の誤入力に対しても対応できる，範囲検索を基にした手法である．

利用者がサーバへ送信する情報を秘匿しつつ，サーバから対応する情報を取得する枠組みとしては，検索可能暗号だけではなく PIR も提案されている [1]．この PIR に関する既存研究は，結託しないことを仮定した複数のサーバを利用するものと単一のサーバのみを用いるもの [2] などが提案されており，プライバシを考慮した位置情報データベースの実現など幅広く用いられている [3]．その中で，クラウド型 IME に適しているものは，単一サーバのみを用い計算量を基にした安全性を保証するもので，cPIR(Computational Private Information Retreval)[4] と呼ばれる．cPIR におけるサーバ計算量は，サーバが受け付ける入力文字列の総数を n として $O(n)$ であり，少ないとは言えない．この サーバにおける計算時間問題を改善する手法として bbPIR [5] がある．これは，cPIR に k-匿名

[1] 九州大学大学院システム情報科学研究院
Faculty of Information Science and Electrical Engineering, Kyushu University, 744 Motooka, Nishi-ku, Fukuoka 819-0395, Japan
[a)] kawamoto@inf.kyushu-u.ac.jp
[b)] sakurai@csce.kyushu-u.ac.jp

[*1] http://www.nikkei.com/article/DGXNASDG2600W_W3A221C1CC0000/

性 [6] の概念を加えたもので，利用者は問合せ前に k 個の入力文字列を含む Bounding box をサーバに送信する．以後は，この k 個の入力文字列のみを受け付けるサブ IME に対して，cPIR プロトコルを実行する．従って，この手法では利用者による入力文字列が k 個のうちどれであるかは秘匿することができるが，元々の cPIR に比べて安全性は下がる．特に，頻出入力文字列が攻撃者に既知の場合，単純に大きさ k の Bounding box を用いると，入力文字列が推測可能である．この対策として，我々は頻度分布を用いた計算量緩和手法を提案している [7].

一方，これらの技術では，入力と出力が一対一の関係であることを前提としており，利用者の誤入力に対応した IME を作成する場合，あらゆる誤入力の組み合わせを予め定義する必要があり現実的ではない．本研究では，入力文字列をある空間に射影する．誤入力文字列を本来の入力文字列の近くに射影し，範囲検索を利用することで，誤入力に対応した IME を実現する．この方法では，あらゆる誤入力の組み合わせを定義する必要はなく，記憶容量の面で効率的である．そのうえで，入力文字列をサーバから秘匿するため，範囲検索可能暗号を利用している．

2. クラウド型 IME のモデル化

本節では，クラウド型 IME をモデル化しその安全性について述べる．

2.1 IME (Input Method Editor)

アルファベット集合を $\Sigma = \{a_1, a_2, \cdots, a_N\}$ と書く．全部で N 種類のアルファベットがある．IME にて変換を行う最大の文字列長を L とする．アルファベット集合に空文字 ϵ を加えた集合を $\Sigma^+ = \Sigma \cup \{\epsilon\}$ とする．

文字列を $\mathbf{w}$ と書き，IME が扱う文字列集合を W と書く．文字列 $\mathbf{w}$ の長さが ℓ とすると，

$$\mathbf{w} = w_1 w_2 \cdots w_\ell \quad (\forall i, w_i \in \Sigma)$$

と書ける．$\ell < L$ の時，$\mathbf{w}$ の後ろに空文字を $L-\ell$ 個連結することで長さ L の正規化された文字列を作成することができる．

$$\tilde{\mathbf{w}} = \mathbf{w}\underbrace{\epsilon\epsilon\cdots\epsilon}_{L-\ell}$$

また，正規化された文字列集合を $\tilde{W}$ を書く．

正規化された文字列に対して順序を定義する．まず，空文字を含むアルファベット集合に番号を定める．$\mathrm{ord} : \Sigma^+ \to \mathbb{Z}^+$ を

$$\mathrm{ord}(a) = \begin{cases} 0 & (a = \epsilon), \\ i & (a = a_i) \end{cases} \tag{1}$$

これを用いて，正規化された文字列 $\tilde{\mathbf{w}} = w_1 w_2 \cdots w_L$ のインデックス $\mathrm{index} : \tilde{W} \to \mathbb{N}$ を次のように定める．

定義 2.1 (正規化された文字列への索引) ある正規化された文字列 $\tilde{\mathbf{w}}$ が $\tilde{\mathbf{w}} = w_1 w_2 \cdots w_L$ であるとき，この文字列 $\tilde{\mathbf{w}}$ の索引を

$$\mathrm{index}(\tilde{\mathbf{w}}) = \sum_{i=1}^{L} \mathrm{ord}(w_i) N^{L-i} \tag{2}$$

と定める．

この索引を用いて隣接語を定義する．

定義 2.2 (δ-近傍語) ある二つの正規化された文字列 $\tilde{\mathbf{w}}$, $\tilde{\mathbf{w}}'$ が，

$$\mathrm{index}(\tilde{\mathbf{w}}) - \mathrm{index}(\tilde{\mathbf{w}}') = \delta$$

を満たすとき，二つの文字列は δ 隣接関係にあるといい，

$$\tilde{\mathbf{w}} = \tilde{\mathbf{w}}' + \delta, \quad \tilde{\mathbf{w}}' = \tilde{\mathbf{w}} - \delta$$

と書く．

本論文で扱う IME とは，利用者から文字列 $\mathbf{w}$ を受け取り，その文字列に対応する変換後の文字列集合を返却するものとする．以降文字列は長さ L に正規化されているものとする．同音異義語が存在するため，各文字列に対応する返還後の文字列は一種類とは限らない．ある正規化された文字列 $\tilde{\mathbf{w}}$ の変換候補となる変換後文字列集合を $R(\tilde{\mathbf{w}})$ と書くことにする．このとき，簡単な IME 次のようなシステムとして定義できる．

定義 2.3 (簡単な IME) 簡単な IME とは，入力として正規化された文字列 $\tilde{\mathbf{w}}$ を受け取り，候補となる変換後文字列集合 $R(\tilde{\mathbf{w}})$ を返すシステムをいう．

製品として提供される IME は，より複雑な機能を提供している．その一つが入力ミスへの対応である．また，入力ミスなどにより本来の文字列と一部異なった文字列を入力し変換を行うこともある．この入力ミスを考慮した変換を行うため，先ほど定義した文字列の順番を利用し，対象となる文字列の前後 δ に含まれる文字列に対する変換候補も合わせて返すことを考える．すなわち，本論で考える IME を次のように定める．

定義 2.4 (δ 入力ミスを許容する IME) δ 入力ミスを許容する IME とは，正規化された文字列 $\tilde{\mathbf{w}}$ を受け取り，候補となる変換後文字列集合として

$$R_\delta(\tilde{\mathbf{w}}) = \bigcup_{\mathbf{w} \in [\tilde{\mathbf{w}}-\delta, \tilde{\mathbf{w}}+\delta]} R(\mathbf{w}) \tag{3}$$

を返すシステムをいう．

このように定めた入力ミスを許容する IME をクラウドサービスとして提供し利用者の入力文字列を秘匿したままサービスを提供するためには，範囲検索を実現する検索可能暗号が必要となる．次の節では，この範囲検索のための検索可能暗号を定義する．

2.2 範囲検索可能暗号

本節では，本研究で用いる暗号化範囲問合せのフレームワークについて述べる．クラウド IME サービス提供者 *SP* は IME のための辞書データをクラウドサービス上のサーバ *Server* に保管する．IME 利用者 *User* は，Server に入力文字列を送信し変換結果候補を取得する．しかし，User は SP と Server を完全には信用しておらず，Server に送信する文字列を暗号化することを考える．暗号化範囲問合せは，User が送信した暗号化された入力文字列を Server 上で復号することなく対応する変換結果候補を得る枠組みである．

SP は Server にデータ t の集合 D を預けるとする．ここで，個々のデータは検索用のキー属性値 k とキー属性値に紐付けられたデータ v の組 $t = (k, v)$ という形をしているとし，キー属性値の定義域は自然数 $\mathbb{N}$ とする．また，関数 key, value を用いて，タプルのキー属性値 k，データ v を表すことにする．すなわち，$\mathrm{key}(t) = k$, $\mathrm{value}(t) = v$ である．データに対する検索は，キー属性の値 k が $[\alpha, \beta]$ $(\alpha \leq \beta \in \mathbb{N})$ に含まれるデータを検索する範囲問合せのみを考える．すなわち，範囲問合せ $q_D(\alpha, \beta)$ は，

$$q_D(\alpha, \beta) = \{t \in D \mid \alpha \leq \mathrm{key}(t) \leq \beta\}$$

と表すことができる．

この暗号化範囲問合せは次の四つのアルゴリズム，鍵生成，属性値暗号化，問合せ生成そして範囲検索からなる．

定義 2.5 (鍵生成) サービス提供者 SP は，User と共有する利用者用秘密鍵 sk_c と Server に渡すサーバ用秘密鍵 sk_s を生成する．鍵生成アルゴリズム KeyGen は，セキュリティパラメータ n を引数とし，秘密鍵のペアを出力する．

$$(sk_c, sk_s) \leftarrow \mathrm{KeyGen}(n)$$

なお，Server は受け取った秘密鍵を安全に管理するものとする．

定義 2.6 (暗号化) SP は，Server に預けるデータ $t = (k, v)$ のキー属性値 $k \in \mathbb{N}$ を暗号化キーベクトル $\mathbf{k_e}$ に変換する．キー属性値暗号化アルゴリズム SERQ は，平文キー属性値 k と利用者用の秘密鍵 sk_c を入力とし，暗号化キーベクトル $\mathbf{k_e}$ を出力する．

$$\mathbf{k_e} \leftarrow \mathrm{SERQ}(k, sk_c)$$

SP は，$(\mathbf{k_e}, v)$ を Server へ送信する．

定義 2.7 (問合せ生成) User は，検索範囲 $[\alpha, \beta] (\alpha \leq \beta \in \mathbb{N})$ を変換した暗号化問合せベクトル $\mathbf{q_e}$ と付加情報 x を生成する．問合せ変換アルゴリズム Query は，問合せ範囲 $[\alpha, \beta]$ と利用者用秘密鍵 sk_c を入力とし，暗号化問合せベクトル $\mathbf{q_e}$ と x の組を出力する．

$$(\mathbf{q_e}, x) \leftarrow \mathrm{Query}([\alpha, \beta], sk_c)$$

User は，得られた $(\mathbf{q_e}, x)$ を Server へ送信する．

定義 2.8 (範囲検索) Server は，User から問合せを受け取ると，保管しているデータ $(\mathbf{k_e}_i, v_i)$ $(i = 1, 2, \cdots)$ それぞれが問合せに該当するか否かをアルゴリズム Test を用いて調べる．判定アルゴリズム Test は，暗号化キーベクトル $\mathbf{k_e}_i$ と問合せ $(\mathbf{q_e}, x)$，サーバ用の秘密鍵 sk_s を入力とし，$res_i \in \{0, 1\}$ を出力する．

$$res_i \leftarrow \mathrm{Test}(\mathbf{k_e}_i, \mathbf{q_e}, x, sk_s)$$

判定結果 res が 1 であった i に対応する v_i の集合を作成し，問合せ元である User に送信する．

2.3 安全な IME

図 1 は，2.2 節で定義した暗号化範囲検索フレームワークを図示したものである．本説では，このフレームワークを用いて利用者のプライバシを考慮したクラウド IME システムを設計する．

クラウド IME のサービスプロバイダは，正規化された入力文字列 $\tilde{\mathbf{w}}$ に対する変換候補集合 $R(\tilde{\mathbf{w}})$ を用意しているものとする．これは，通常 IME のための辞書作成と同様である．これを $t = (\tilde{\mathbf{w}}, R(\tilde{\mathbf{w}})$ $(\forall w \in W)$ としてサーバに保存することにする．暗号化範囲検索フレームワークを用いると，SP は秘密鍵 sk_c とアルゴリズム SERQ を用いて，キー属性値は自然数である必要がある．そこで，式 (2) にて定義した索引を用いて，暗号化キーベクトルとして，

$$\mathbf{k_e} = \mathrm{SERQ}(\mathrm{index}(\tilde{\mathbf{w}}), sk_c)$$

を用いる．従って，SP はすべての $\tilde{w} \in \tilde{W}$ に対して，$(\mathrm{SERQ}(\mathrm{index}(\tilde{\mathbf{w}}), sk_c), R(\tilde{\mathbf{w}}))$ を計算し，サーバへ送信する．

利用者が文字列 $\mathbf{w}$ を入力し，クラウド IME を用いて変換を行う場合，クラウド IME のクライアントは，入力文字列を正規化しアルゴリズム Query を用いて問合せを作成する．なお，このクラウド IME は δ を与えられたパラメータとして，δ 入力ミスを許容する IME であるとする．この時，クライアントは，

$$(\mathbf{q_e}, x) = \mathrm{Query}(\mathrm{index}(\tilde{\mathbf{w}}) - \delta, \mathrm{index}(\tilde{\mathbf{w}}) + \delta, sk_c)$$

によって問合せを作成しサーバへ送信する．

サーバは，受け取った問合せをもとに，すべての $t \in D$ に対して，アルゴリズム Test を実行し，Test の結果が 1 であるタプル t のデータ部分の和集合を取り利用者へ返却する．アルゴリズム 1 は，このサーバの処理を表したものである．これまでの議論から，サーバから返却される Res には，利用者が入力した文字列 $\tilde{\mathbf{w}}$ に対し，$[\tilde{\mathbf{w}} - \delta, \tilde{\mathbf{w}} + \delta]$ に含まれる変換候補が含まれていることがわかる．すなわち，式 (3) を満たす変換候補集合 $R_\delta(\tilde{\mathbf{w}})$ の取得が行えていることがわかる．

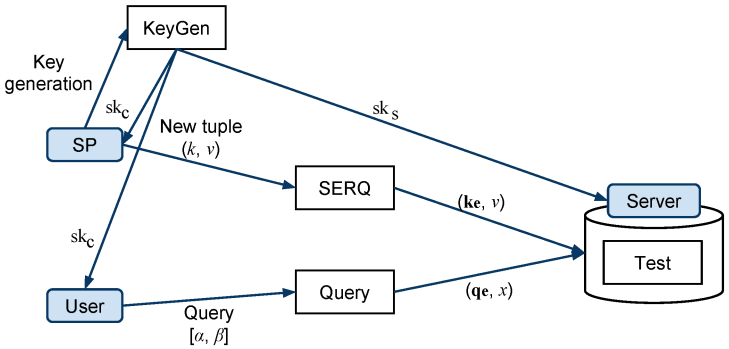

図 1 範囲検索可能暗号のフレームワーク.

Algorithm 1 安全な IME におけるサーバの動作.

Require: Query $(\mathbf{q_e}, x)$
Require: Set of candidate data D
Require: Encryption key of the server sk_s
1: $Res \leftarrow \emptyset$
2: **for** each tuple t in D **do**
3: $\quad \mathbf{k_e} \leftarrow \mathrm{key}(t)$
4: $\quad$ **if** $\mathrm{Test}(\mathbf{k_e}, \mathbf{q_e}, x, sk_s) = 1$ **then**
5: $\quad\quad Res \leftarrow Res \cup \mathrm{value}(t)$
6: $\quad$ **end if**
7: **end for**
8: **return** Res

3. 内積述語による範囲検索可能暗号

本節では，範囲検索可能暗号の一つとして IPP 法 [8] を説明する．この方式ではキー属性値と範囲問合せの両方に乱数を加え暗号化を施すことで内容の漏えいを防ぐ．本論文では，この方式をクラウド型 IME へ利用する．

IPP 法では，キー属性値の漏えいを防ぐためにキー属性値に摂動を加えている．摂動には $0 < r_k < 1/2$ を満足する乱数 r_k を用い，キー属性値 k に摂動を加えた値を $k + r_k$ とする．キー属性値に摂動を加える操作を

$$\mathrm{per} : \mathbb{N} \times \mathbb{R} \to \mathbb{R},\ \mathrm{per}(x; r_k) = x + r_k$$

と表す．なお，摂動 r_k は毎回異なる値をランダムに選ばれる．つまり同じキー属性値であっても別の値が使用される．

キー属性値に摂動を加えたタプルからなるデータベースを摂動を加えたデータベース PD と呼ぶことにする．この時，元々のタプル集合 D と摂動を加えたデータベース PD の関係は，

$$PD = \{(\mathrm{per}(\mathrm{key}(t); r_k \underset{r}{\leftarrow} [0, \frac{1}{2}]), \mathrm{value}(t)) \mid \forall t \in D\}$$

である．ただし，$r_k \underset{r}{\leftarrow} [0, 1/2]$ は，タプル毎に異なるランダムな値を $[0, 1/2]$ から選ぶことを意味する．摂動を加えたデータベースに対する問合せ q_{PD} は，キー属性の定義域が自然数かつ $0 < r_k < 1/2$ という条件から次のように変換できる．すなわち，区間 $[\alpha, \beta]$ を問い合わせる問合せは

$$q_{PD}(\alpha, \beta) = \{t \in PD | \alpha - \frac{1}{2} \le \mathrm{key}(t) \le \beta + \frac{1}{2}\}$$

となる．

3.1 多項式述語

問合せに対して摂動を加えるために，我々は問合せを内積述語という形で表現する．ここでは，先ず内積述語の元となる多項式述語を導入する．

多項式述語では，範囲問合せを多項式 $p : \mathbb{N} \to \mathbb{R}$ を用いて表現する．述語の値域は実数全体であるが，次の様に定める sign 関数

$$\mathrm{sign}(x) = \begin{cases} 1 & (x \ge 0) \\ 0 & (\mathrm{otherwise}) \end{cases}$$

を用い，$\mathrm{sign} \circ p$ とすることで 2.2 節で定義した Test 関数を実現する．

区間 $[\alpha, \beta]$ の範囲問合せを表す多項式述語 $p_{[\alpha,\beta]}$ は複数存在するが，我々は，最も単純な多項式の一つである

$$p_{[\alpha,\beta]}(k) = -(k - \alpha)(k - \beta)$$

を用いる．この多項式に対して，$\mathrm{sign} \circ p_{[\alpha,\beta]}(k)$ は $k \in [\alpha, \beta]$ の時に限り 1 となることは自明である．

我々が対象としているデータベース PD では，キー属性値には摂動 r_k が加えられている．この摂動付きキー属性値に対し，偽陰性を含まず問合せ処理を行うために多項式述語のパラメータに摂動による増加分を加える必要がある．具体的には，区間 $[\alpha, \beta]$ の範囲問合せを表す先ほどの多項式述語は，

$$p'_{[\alpha,\beta]}(k) = -(k - \alpha + \frac{1}{2})(k - \beta - \frac{1}{2})$$

となる．

次に，この多項式述語に摂動を加える．我々は，摂動に正の乱数 r_q を用い，多項式述語に $(k + r_q)$ を乗じることで摂動を加える．先ほどの多項式述語 $p'_{[\alpha,\beta]}$ の場合，摂動を加えた述語 $\tilde{p}'_{[\alpha,\beta];r_q}(k)$ は

$$\tilde{p}'_{[\alpha,\beta];r_q}(k) = \mathrm{per}(p'_{[\alpha,\beta]}(k); r_q) = -(k-\alpha+\frac{1}{2})(k-\beta-\frac{1}{2})(k+r_q) \quad (4)$$

となる．キー属性値の定義域は自然数と仮定しているので，$k \in [\alpha, \beta]$ であれば (4) はゼロ以上となる．すなわち，$\mathrm{sign} \circ \tilde{p}'_{[\alpha,\beta];r_q}(k)$ は $k \in [\alpha, \beta]$ の時に限り 1 となる．

3.2 内積述語

先ほど定義した多項式述語をベクトルの内積を用いた述語である内積述語として表現する．一般に，任意の一変数多項式は定数ベクトルと変数ベクトルの内積として表すことができる．この性質により，(4) の摂動を加えた多項式述語は，定数ベクトル

$$\mathbf{q}_{[\alpha,\beta];r_q} = \begin{pmatrix} -1 \\ \alpha+\beta-r_q \\ -(\alpha-\frac{1}{2})(\beta+\frac{1}{2}) + (\alpha+\beta)r_q \\ -(\alpha-\frac{1}{2})(\beta+\frac{1}{2})r_q \end{pmatrix}^t \quad (5)$$

と変数ベクトル $\mathbf{k} = (k^3, k^2, k, 1)^t$ の内積 $\mathbf{q}_{[\alpha,\beta];r_q} \cdot \mathbf{k}$ と表すことができる[*2]．

このことを用いて，キー属性値が k であるタプルには，k の代わりにベクトル化した $(k^3, k^2, k, 1)^t$ をキー属性値として与えて置けば，範囲問合せ $[\alpha, \beta]$ に対応する内積述語の一つは

$$\mathrm{IPP}_{[\alpha,\beta],r_q}(\mathbf{k}) = \mathbf{q}_{[\alpha,\beta],r_q} \cdot \mathbf{k}$$

となる．実際には，キー属性値 k には摂動が追加され，$\mathrm{per}(k, r_k) = k + r_k$ となっている．したがって，キー属性値が k であるタプルには k の代わりに摂動付きキーベクトル

$$\mathbf{k}' = (\mathrm{per}(k; r_k)^3, \mathrm{per}(k; r_k)^2, \mathrm{per}(k; r_k), 1)^t \quad (6)$$

が与えられることになる．前節で議論したように，摂動付きキー属性値に対しても，キー属性値 k が $k \in [\alpha, \beta]$ であれば，多項式述語 $\tilde{p}'_{[\alpha,\beta],r_q}$ がゼロ以上の値となる．したがって，多項式述語の等価な変形である内積述語 $\mathrm{IPP}_{[\alpha,\beta];r_q}$ もゼロ以上の値となり，$\mathrm{sign} \circ \mathrm{IPP}_{[\alpha,\beta];r_q}(\mathbf{k})$ は，$k \in [\alpha, \beta]$ の時に限り 1 となる．

4. 巡回行列による範囲検索可能暗号

IPP 法 [8] では，キーベクトル $\vec{k'}$ と問合せに用いるベクトル $\vec{q}_{[\alpha,\beta];r_q}$ の値を秘匿するために，正則行列 M を用いて，$M^{-1}\vec{k'}$ と $M^t\vec{q}_{[\alpha,\beta];r_q}$ をそれぞれ暗号化されたキーベクトルと問合せに用いるベクトルとして用いていた．このとき，二つのベクトルの内積は，

$$M^t\vec{q}_{[\alpha,\beta];r_q} \cdot M^{-1}\vec{k'} = \vec{q}^t_{[\alpha,\beta];r_q} M M^{-1} \vec{k'} = \vec{q}_{[\alpha,\beta];r_q} \cdot \vec{k'}$$

となり，内積結果は暗号化の前後で変化しない．

一方，この方式では，キーベクトルと問合せを盗み見れる攻撃者はそのキーベクトルが問合せに合致するのか確認することが出来る．本論文では，Bob にもサーバ用の鍵を渡すことで，サーバになりすます第三者からの攻撃に耐性を持つ方法へ拡張する．

4.1 巡回行列による暗号

本提案手法では，暗号鍵として n 巡回行列を利用する．n 巡回行列 A とは $A^n = E$ かつ $A^i \neq E \quad (i < n)$ を満たす行列 A のことを言う．なお，E は単位行列である．この巡回行列 A を用いて，利用者のための鍵を $sk_c = (A, n, c, s)$ とする．ここで，c と s は n 未満の乱数である．そして，サーバ用の鍵として $sk_s = A^s$ を用いる．

これらの鍵を用いた場合のキーベクトルの暗号化方法は次の通りである．暗号化前のキーベクトルが $\mathbf{k}'$ であるとすると，乱数 r_e を用いて，$r_e A^c \mathbf{k}'$ を暗号化したキーベクトル $\mathbf{k_e}$ とする．また，問合せ用のベクトルについては，乱数 r を用いて $(A^r)^t\mathbf{q}$ を暗号化した問合せベクトル $\mathbf{q_e}$ とする．さらに，問合せ時には，x についての方程式

$$sx + c + r = 0 \pmod{n} \quad (7)$$

を解き，得られた x を暗号化した問合せベクトルと併せてサーバに送る．

定理 4.1 (暗号化ベクトルに対する内積) サーバは自信が持つサーバ用の鍵 A^s を用いて，$\mathbf{q_e} \cdot (A^s)^x \mathbf{k_e}$ を計算することで暗号化前のベクトルに対する内積を計算できる．

証明 4.1

$$\begin{aligned} \mathbf{q_e} \cdot (A^s)^x \mathbf{k_e} &= (A^r)^t\mathbf{q} \cdot (A^s)^x r_e A^c \mathbf{k}' \\ &= r_e \mathbf{q}^t A^r A^{sx} A^c \mathbf{k}' \\ &= r_e \mathbf{q}^t A^{sx+c+r} \mathbf{k}' \\ &= r_e \mathbf{q}^t \mathbf{k}' = \mathbf{q} \cdot \mathbf{k}' \end{aligned}$$

従って，

$$\mathrm{sign}(\mathbf{q_e} \cdot (A^s)^x \mathbf{k_e}) \quad (8)$$

を判定すればその暗号化キーベクトルを持つタプルが問合せ内容に合致しているか否かを判定できる．また，サーバ用の暗号鍵 A^s を知らない攻撃者は問合せと暗号化タプルを入手したとしても問合せに合致するタプルか否かの判定はできない．

4.2 SERQ アルゴリズム

SP は始めに，アルゴリズム KeyGen を用いて利用者用の秘密鍵 sk_c とサーバ用の秘密鍵 sk_s を生成する．この手続をアルゴリズム 2 に示す．KeyGen は，与えられたセキュリティパラメータ n を元に 4 次正方 n 巡回行列と互いに素な整数 c, s を生成する．それらを用いて二つの鍵を

[*2] $\mathbf{x}^t$ でベクトル $\mathbf{x}$ の転値ベクトルを，$\mathbf{x} \cdot \mathbf{y}$ でベクトル $\mathbf{x}$ と $\mathbf{y}$ の内積を表す．

Algorithm 2 KeyGen

Require: Big prime parameter n
1: Generate n-ordered cyclic matrix A
2: Choose $c, s < n$, where c and s are relatively prime
3: $sk_c \leftarrow (A, n, c, s)$, $sk_s \leftarrow A^s$
4: **return** (sk_c, sk_s)

Algorithm 3 SERQ

Require: Attribute value $k \in \mathbb{N}$
Require: Encryption key of clients $sk_c = (A, n, c, s)$
1: $r_k \xleftarrow{r} [0, 1/2]$
2: $\mathbf{k}' = (\mathrm{per}(k; r_k)^3, \mathrm{per}(k; r_k)^2, \mathrm{per}(k; r_k), 1)^t$
3: $r_e \xleftarrow{r} R^+$
4: $\mathbf{k_e} \leftarrow r_e A^c \mathbf{k}'$
5: **return** $\mathbf{k_e}$

Algorithm 4 Query

Require: Querying range $[\alpha, \beta] (\alpha \leq \beta \in D_K)$
Require: Encryption key of clients $sk_c = (A, n, c, s)$
1: Compute $\mathbf{q}$ by (5)
2: $r \xleftarrow{r} \mathbb{Z}$
3: $\mathbf{q_e} \leftarrow (A^r)^t \mathbf{q}$
4: Solve for x: $sx + c + r = 0 \pmod n$
5: **return** $(\mathbf{q_e}, x)$

計算し出力する．なお，3 次正方 n 巡回行列を生成するひとつの方法を付録に記している．SP は，作成した利用者用の秘密鍵 sk_c を User に，サーバ用の秘密鍵 sk_s を Server に渡す．

SP が Server に新しいデータ (k, v) を預けたいとする．この時，SP は，アルゴリズム SERQ を用いてキー属性値 k の代わりに用いる暗号化キーベクトル $\mathbf{k_e}$ を求める．この手続をアルゴリズム 3 に示す．SERQ は，先ず，$[0, 1/2]$ 内の乱数 r_k を生成し 6 で導入したキーベクトル $\mathbf{k}'$ を計算する．その後，新たに生成した正の乱数 r_e と利用者用の秘密鍵 sk_c の中から A と c を用いて暗号化キーベクトルを計算する．SP は，SERQ によって得られた暗号化キーベクトルを用いたデータ $(\mathbf{k_e}, v)$ を Server へ送信する．なお，タプル中の v は一般的な暗号化手法を用いて暗号化されているものとする．

User が Server からキー属性値 k が区間 $[\alpha, \beta]$ に含まれるデータを取得したいとする．この時，User はアルゴリズム Query を用いて問合せを作成する．この手続をアルゴリズム 4 に示す．Query は，(5) で導入した問合せベクトルを計算する．そして，整数乱数 r を生成し利用者用秘密鍵 sk_c を用いて暗号化問合せベクトル $\mathbf{q_e}$ を計算する．次に，x についての方程式 (7) を解き x を得る．最後に，それらの組 $(\mathbf{q_e}, x)$ を出力する．User は Query によって得られた問合せを Server へ送る．

Server は，User から問合せ $(\mathbf{q_e}, x)$ を受け取ると，保持しているデータすべてに対しアルゴリズム Test を適用

Algorithm 5 Test

Require: Encrypted key vector $\mathbf{k}_{\mathbf{e}i}$
Require: Query $(\mathbf{q_e}, x)$
Require: Secret key of the server $sk_s = A^s$
1: **return** $\mathrm{sign}(\mathbf{q_e} \cdot (A^s)^x \mathbf{k}_{\mathbf{e}i})$

Algorithm 6 Encrypted Range Search with Query

Require: Query $(\mathbf{q_e}, x)$
Require: Database PD
Require: Encryption key of the server $sk_s = A^s$
1: $Res \leftarrow \emptyset$
2: **for** each tuple t in PD **do**
3: $\quad \mathbf{k_e} \leftarrow \mathrm{key}(t)$
4: $\quad$ **if** $\mathrm{Test}(\mathbf{k_e}, \mathbf{q_e}, x, sk_s) = 1$ **then**
5: $\quad\quad Res \leftarrow Res \cup \{\mathrm{value}(t)\}$
6: $\quad$ **end if**
7: **end for**
8: **return** Res

し問合せに合致するタプルか否かを判定する．この Test は，アルゴリズム 5 に示すとおりで，式 (8) を判定する．Server は，Test が 1 を返すタプルを集め，User へ送信する．この手続をアルゴリズム 6 に示す．

以上のアルゴリズムを 2.3 節の議論と合わせることで，利用者の入力文字列を秘匿しながら変換文字列を取得する，利用者のプライバシを考慮したクラウド IME を実現することができる．

5. まとめと今後の課題

本論文では，近年話題となっているクラウド型アプリケーションにおけるプライバシ保護技術，中でもクラウド型 IME におけるプライバシ保護技術を提案した．我々が提案するクラウド型 IME は単純な文字列変換だけではなく，入力ミスへも対応できる．このクラウド型 IME を実現するため，範囲検索可能暗号を利用し，内積計算と循環行列を用いた一つの実現方法を用いた．今後の課題としては，本提案手法を実装し，計算コストや実際の文字列処理に係る時間の計測を考えている．

謝辞 本研究は JSPS 科研費 23300027 及び 26730065, 財団法人人工知能研究振興財団の支援を受けたものである．ここに記して謝意を表します．

参考文献

[1] Chor, B., Goldreich, O., Kushilevitz, E. and Sudan, M.: Private Information Retrieval, *Journal of the ACM*, Vol. 45, No. 6, pp. 965–982 (1998).
[2] Ostrovsky, R. and Skeith, III, W. E.: A Survey of Single-Database PIR: Techniques and Applications, *Proc. of the 10th International Conference on Practice and Theory in Public-key Cryptography*, Springer, pp. 393–411 (2007).
[3] Ghinita, G., Kalnis, P., Khoshgozaran, A., Shahabi, C. and Tan, K.-L.: Private Queries in Location Based Ser-

vices: Anonymizers are Not Necessary, *Proc. of the 28th ACM SIGMOD International Conference on Management of Data*, ACM Press, pp. 121–132 (2008).

[4] Kushilevitz, E. and Ostrovsky, R.: Replication Is Not Needed: Single Database, Computationally-Private Information Retrieval, *Proc. of the 38th Annual Symposium on Foundations of Computer Science*, IEEE Computer Society, pp. 364–373 (1997).

[5] Wang, S., Agrawal, D. and Abbadi, A. E.: Generalizing PIR for Practical Private Retrieval of Public Data, *Proc. of the 24th Annual IFIP WG 11.3 Working Conference on Data and Applications Security and Privacy*, Springer, pp. 1–16 (2010).

[6] Sweeney, L.: k-Anonymity: A Model for Protecting Privacy, *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, Vol. 10, No. 5, pp. 1–14 (2002).

[7] Kawamoto, J. and Gillett, P. L.: Frequency-based Constraint Relaxation for Private Query Processing in Cloud Databases, *Proc. of the 27th Annual IEEE Canadian Conference on Electrical and Computer Engineering*, IEEE Computer Society (2014).

[8] Kawamoto, J. and Yoshikawa, M.: Private Range Query by Perturbation and Matrix Based Encryption, *Proc. of the Sixth IEEE International Conference on Digital Information Management*, IEEE Computer Society, pp. 211–216 (2011).