「マルチメディア，分散，協調とモバイル
(DICOMO2017) シンポジウム」 平成29年6月

秘匿検索の頻度分析対策としての複数DB活用について

伊藤 隆[1]　平野 貴人[1]　森 拓海[1]　服部 充洋[1]

概要：秘匿検索（検索可能暗号）は，ユーザがデータセンタに登録した暗号化データを，暗号化したままで検索するための技術である．秘匿検索では，暗号化キーワードの頻度からキーワードを推定する頻度分析攻撃に対抗するため，確率的暗号を用いるなどして頻度情報を隠ぺいする．しかし，秘匿検索システムが運用されて検索が行われると，キーワードの同一性や頻度情報が徐々に漏れるため，最終的には頻度分析攻撃が可能になるという問題があった．本論文では，暗号化データを複数データセンタのいずれかに振り分け，この際各データセンタにおける頻度分布が撹乱されるよう工夫することで，効率面でのオーバーヘッドを生じることなく，頻度分析対策を実現する方式を提案する．また，提案方式による頻度分析耐性の向上を，攻撃者によるキーワード推定の的中率に基づいて評価した結果，都道府県名を秘匿する例で的中率を 78%から 17%に低減できるなど，安定した効果を持つことが確認できた．

Introducing Multiple Databases in Searchable Encryption as a Countermeasure to the Frequency Analysis

TAKASHI ITO[1]　TAKATO HIRANO[1]　TAKUMI MORI[1]　MITSUHIRO HATTORI[1]

1. はじめに

秘匿検索（検索可能暗号）は，ユーザがデータセンタに登録した暗号化データを，暗号化したままで検索するための技術である．秘匿検索を用いると，データセンタへの機密情報のアウトソーシングや，データセンタ上での暗号化メールのフィルタリングなどが実現可能となるため，機能性・安全性・効率性などを考慮した，様々な秘匿検索方式が提案されている [1], [2], [3], [4], [5], [6], [7], [8].

検索用のキーワードを直接暗号化するタイプの秘匿検索の場合，同じキーワードが同じ暗号化キーワードになることで頻度分析攻撃されることを防ぐため，通常はキーワードを確率的暗号で暗号化する [1], [2]．一方で，確定的暗号をベースとしつつ，頻度分析攻撃への耐性を持たせた方式も提案されている [6], [7].

しかし，確率的暗号・確定的暗号のいずれを用いる場合でも，実際に秘匿検索システムが運用されて検索が行われると，検索要求とマッチした複数の暗号化キーワードについてキーワードの同一性が判明し，その頻度情報もデータセンタに漏れてしまう．これが繰り返されることで，最終的にはデータセンタによる頻度分析攻撃が可能になるという問題があった．

この問題に対し，山岡らの論文 [8] では，全キーワードを，k (≥ 2) 個以上のキーワードで構成されるグループのいずれかに分類し，検索時にはキーワードの代わりにグループ ID を用いる方式が提案されている．この方式の利点は，攻撃者がグループ ID や，グループを構成するキーワード群を特定できた場合でも，正しいキーワードが k 個のうちどれであるかを特定できないことである．しかしこの方式には，検索処理量，特にデータセンタからデータ検索者に送信される検索結果のサイズが約 k 倍に悪化するという，効率面での課題がある．

そこで本論文では，暗号化データを複数データセンタのいずれかに振り分け，この際各データセンタにおけるキーワードの頻度分布が，元のキーワードの頻度分布から変化するよう工夫することで，攻撃者による頻度分析攻撃を困難にする方式を提案する．提案方式では，キーワードごとにあらかじめ設定した振り分け確率に従い，登録先データセンタをランダムに選択する処理が本質であり，既存の多くの秘匿検索方式への適用が期待できるとともに，データセンタのストレージや，各エンティティの計算量・通信量

[1] 三菱電機株式会社 情報技術総合研究所

を増加させることなく，秘匿検索の頻度分析対策を実現することが可能である．

併せて，提案方式による頻度分析耐性の向上を，合理的な攻撃者によるキーワード推定の的中率を評価指標として，定量的に評価した結果を示す．具体的には，例えば都道府県名を秘匿検索対象として 10 万人分の情報が登録される場合で，無対策の場合に 78%となる的中率を，提案方式の適用で 17%まで低減できるなどの結果が得られた．

2. 既存の秘匿検索方式と課題

本節では，秘匿検索について簡単に説明したのち，秘匿検索システムの分類および既存方式の紹介を行う．その後，秘匿検索システムに対する攻撃の一つである「頻度分析攻撃」について説明する．

2.1 秘匿検索システムのエンティティと要件

秘匿検索（検索可能暗号）技術とは，暗号化データを暗号化したままで検索する技術の総称である．本論文で扱う「秘匿検索システム」では，登場するエンティティは以下の三つとする．

- **データセンタ**：データ検索者が利用するデータを，データ検索者に代わって保持するエンティティ．データ検索者に対して検索機能を提供する．
- **データ登録者**：データセンタにデータを登録するエンティティ．通常，データ検索者がデータを検索できるよう，各データに対して適切なキーワードを付与する．なお，データ登録者と，次のデータ検索者は同一のエンティティが兼ねることもある．
- **データ検索者**：データセンタに登録されたデータを検索するエンティティ．データ検索や，検索後に取得したデータの復号に利用する秘密情報を持っている．

秘匿検索システムでは，データセンタに対して余計な情報を漏らさないまま，データセンタによる検索処理を可能とすることが求められる．具体的には，以下のような要件が設定される．

- **要件 1（機能性）**：データ検索者が，指定した検索キーワードに対応するデータを得られること．
- **要件 2（安全性）**：データセンタに対し，データ検索者が利用するデータの情報，およびデータに付与されたキーワード（登録キーワード）の情報が漏れないこと．
- **要件 3（効率性）**：ストレージ・計算量・通信量などの面で効率的であること．

2.2 秘匿検索システムの分類

本節では，秘匿検索システムをいくつかの観点から分類し，それぞれに対応する関連研究を挙げる．

2.2.1 公開型と非公開型

秘匿検索システムは，データ登録の権限を一般に公開するか，もしくは決められたエンティティのみに限定するかで，以下の二つに大別することができる．

- **公開型**：データ登録の権限を一般に公開するもの．データや検索用のキーワードは公開鍵暗号で暗号化される．外部から来る暗号化メールを，暗号化したまま分類したい場合などに利用できる．代表例は [2], [4] など．
- **非公開型**：データ登録者にも秘密情報を持たせることで，データ登録の権限を，決められたエンティティのみに限定するもの．データやキーワードの暗号化に共通鍵暗号を用いることが多いが，公開鍵暗号を利用しつつ，公開鍵を非公開にするといった使い方も可能である．企業内のデータをデータセンタに外部委託したい場合などに用いられる．代表例は [1], [3], [5] など．

2.2.2 タグ型とインデックス型

- **タグ型**：個々の検索用キーワードから**タグ**と呼ばれる暗号化キーワードを生成し，検索時には各タグと検索要求とのマッチを逐一確認するもの．次のインデックス型に比べ，データの追加・削除が容易であるという長所を持つ．代表例は [1], [2], [4] など．
- **インデックス型**：あらかじめ，文書とそれに含まれるキーワードの対応表である**インデックス**を作成しておき，検索時にはインデックスに対して処理を行うことで検索用キーワードを含む文書を特定する．一般にタグ型よりも高速であるが，データの追加・削除に複雑な処理を必要とすることが多い．代表例は [3], [5] など．

2.2.3 確定的暗号ベースと確率的暗号ベース

特にタグ型の秘匿検索システムについて，キーワードからタグを生成する際に利用する暗号方式によって，以下のような分類も行われる．両方式では，検索時にタグと検索要求とのマッチを確認する方法が異なる．

- **確定的暗号ベースの方式**：キーワードからタグを生成する際に確定的暗号を利用するもの．確定的暗号を用いて単純にタグを生成する場合，同一のキーワードから常に同一のタグが生成されるため，検索時のマッチ確認ではタグと検索要求のバイナリ一致判定を行うだけで良く，レコード数 n に対して $\mathrm{O}(\log n)$ で検索できる利点がある．ただし，そのままでは頻度分析攻撃によってキーワードを推定されるリスクがあるため，通常は何らかの対策を行う．代表例は [4], [6], [7] など．
- **確率的暗号ベースの方式**：キーワードからタグを生成する際に確率的暗号を利用するもの．同一のキーワードから毎回異なるタグが生成されるため，検索時のマッチ確認ではタグと検索要求の間で特殊な演算を行う必要があり，レコード数 n に対して $\mathrm{O}(n)$ の検索時間を必要とする．一方で，多数のタグを観測してもキーワードの頻度情報が漏れないため，確定的暗号ベースの方式よりは安全性が高いとされる．代表例は [1], [2] など．

表 1　暗号化 DB

レコード ID	暗号化データ	タグ
1	$E_1(d_1)$	$E_2(w_1)$
2	$E_1(d_2)$	$E_2(w_2)$
3	$E_1(d_3)$	$E_2(w_3)$
4	$E_1(d_4)$	$E_2(w_4)$
5	$E_1(d_5)$	$E_2(w_5)$
⋮	⋮	⋮

表 2　暗号化 DB（確定的暗号でタグを生成する場合）

レコード ID	暗号化データ	タグ
1	$E_1(d_1)$	E_{D}(北海道)
2	$E_1(d_2)$	E_{D}(東京)
3	$E_1(d_3)$	E_{D}(三重)
4	$E_1(d_4)$	E_{D}(岩手)
5	$E_1(d_5)$	E_{D}(東京)
⋮	⋮	⋮

2.3　本論文で対象とする秘匿検索システム

本論文では，非公開型，かつタグ型の秘匿検索システムを対象とし，下記の手順に従って動作するシステムを想定する．以下ではデータセンタを $\mathcal{C}$，データ登録者を $\mathcal{R}$，データ検索者を $\mathcal{S}$ で表す．また，データ暗号化アルゴリズムを E_1，キーワード暗号化アルゴリズムを E_2 とし，E_2 に対応する検索要求生成アルゴリズムを Q とする．なお，E_2 が確定的・確率的のいずれであるかは問わない．E_2 が確定的である場合，Q は E_2 と同一アルゴリズムとなる．

準備

(1) $\mathcal{S}$ は，データ秘匿用の鍵，キーワード秘匿用の鍵を生成する．また，データ・キーワードの暗号化に用いる鍵を $\mathcal{R}$ に通知する．

登録

(2) $\mathcal{R}$ は，$\mathcal{C}$ に登録するデータ d に対し，登録キーワード w を設定する．簡単のため，ここでは各データに対し，単一の登録キーワードを設定するものとする．

(3) $\mathcal{R}$ は，データ d およびキーワード w を暗号化し，得られた暗号化データ $E_1(d)$ およびタグ $E_2(w)$ を $\mathcal{C}$ に送信する．

(4) $\mathcal{C}$ は，$\mathcal{R}$ から受信した $E_1(d), E_2(w)$ を対応付けたうえで，暗号化 DB に登録する．複数のデータを受信・登録した結果，暗号化 DB は例えば**表 1** のようになる．

検索

(5) $\mathcal{S}$ は，検索キーワード w' に対応する検索要求 $Q(w')$ を生成し，$\mathcal{C}$ に送信する．

(6) $\mathcal{C}$ は，暗号化 DB の各レコードについて，タグ $E_2(w_i)$ が検索要求 $Q(w')$ とマッチするかを確認する．

(7) $\mathcal{C}$ は，マッチしたタグに対応する暗号化データ $E_1(d_i)$ を $\mathcal{S}$ に送信する．

(8) $\mathcal{S}$ は，受信した $E_1(d_i)$ を復号し，データ d_i を得る．

2.4　秘匿検索システムに対する頻度分析攻撃

2.4.1　確定的暗号に対する頻度分析攻撃

前述のように，確定的暗号を利用してタグを生成すると高速な検索が可能という利点があるが，確定的暗号を単純に用いただけでは，観測したタグの頻度分布からタグに対応するキーワードを推定する「頻度分析攻撃」に弱いという欠点がある．

表 3　タグの頻度分布

タグ	出現数（比率）
E_{D}(東京)	10528（0.105）
E_{D}(神奈川)	7393（0.074）
E_{D}(大阪)	6882（0.069）
E_{D}(愛知)	5910（0.059）
E_{D}(埼玉)	5684（0.057）
⋮	⋮
総数	100000

表 4　キーワードの頻度分布

都道府県	人口比率
東京	0.106
神奈川	0.072
大阪	0.070
愛知	0.059
埼玉	0.057
⋮	⋮

例として，日本居住者に関する個人情報 d_i を暗号化して DB に登録する際，検索用のキーワードとして都道府県名を利用し，これを確定的暗号化することを考える．この場合，確定的暗号化アルゴリズムを E_{D} とすると，データセンタが保持する暗号化 DB は例えば**表 2** のようになる．ここで，レコード ID 2,5 におけるタグ E_{D}(東京) は同一の値となることから，データセンタは元のキーワードが同一であることが分かる．さらに，DB に含まれる全タグの出現数を数えることで，例えば**表 3** のような頻度分布を得ることができる（この時点では各タグに対応するキーワードは判明していない）．なお，表 3 のような，特定 DB に含まれるタグの頻度を以降では単に「**タグの頻度**」と記す．

一方，都道府県の人口比率は**表 4**（平成 27 年国勢調査の結果 [9] から算出）のとおりであることが知られているため，データセンタはこれらの頻度を比較することで，各タグに対応する都道府県名を推定することが可能となる．実際，ここで示した例の場合，10 万件のレコードを観測したデータセンタは，78%のレコードについて正しい都道府県名を推定できる（詳しくは 4 節で説明する）．なお，表 4 のような，各キーワードと対応付いたキーワードの母集団頻度を以降では単に「**キーワードの頻度**」と記す．

確定的暗号ベースの方式における頻度分析対策として，検索に使用しない単語に乱数をパディングすることでタグの種類を増加させる方式 [6] や，偏ったキーワードの頻度を分割することで頻度を均一化し，キーワードの特定を困難にする方式 [7] などが提案されている．

2.4.2　検索で判明する頻度を利用した頻度分析攻撃

確率的暗号ベースの方式や，前述の対策を行った確定的暗号ベースの方式では，DB 中のタグのみを観測した頻度分析攻撃は成立しない．しかし，いずれの場合でも，検索

処理が行われ，データセンタが検索要求 $Q(w')$ を受け取ってマッチを確認する（2.3 節の手順（6））と，$Q(w')$ にマッチした全てのタグに関して，対応するキーワードが同一であることが分かり，その出現数も判明する．最も単純な例では，2 値データ（性別，合否，病気の有無など）を秘匿検索対象とした場合，1 回の検索が行われるだけでキーワードの同一性と出現数が全て判明するため，頻度分析攻撃が可能となる．また，都道府県名のようにキーワードの種類が多い場合でも，システムで十分な検索が行われると，全てのタグに関する出現数が判明すると考えられるため，最終的にはデータセンタによる頻度分析攻撃が可能になってしまう．なお，検索後に判明した同一性に基づくタグの頻度に関しても，以降では単に「タグの頻度」と記す．

単純な回避策として，実在するキーワードや架空のキーワードから生成したタグを含む，ダミーのレコードを追加することでタグの頻度を撹乱する方法が考えられるが，ダミー追加に伴うストレージ・計算量・通信量のオーバーヘッドが避けられないうえ，頻度分析耐性が確実に向上するようなダミー追加の方法は自明でない．

一方，ダミーのレコードを使わない方法として，山岡らの論文 [8] では，全キーワードを，k (≥ 2) 個以上のキーワードで構成されるグループのいずれかに分類し，検索時にはキーワードの代わりにグループ ID を用いる方式が提案されている．この方式の利点は，攻撃者がグループ ID や，グループを構成するキーワード群を特定できた場合でも，正しいキーワードが k 個のうちどれであるかを特定できないことである．しかしこの方式には，検索処理量，特にデータセンタからデータ検索者に送信される検索結果のサイズが約 k 倍に悪化するという，効率面での課題がある．

3. 提案方式

検索で判明する頻度の利用も含めた頻度分析攻撃に対し，既存の対策は，データセンタのストレージや計算量，通信量などに何らかのオーバーヘッドを生じるものであった．本節では，これらのオーバーヘッドを生じない頻度分析対策として，登録データを複数データセンタのいずれかに振り分け，この際各データセンタにおけるタグの頻度が撹乱されるよう工夫することで，頻度分析攻撃を困難にする方式を提案する．

なお，提案方式では複数データセンタを利用するが，各データセンタは結託しないことを前提とする．

3.1 特徴

提案方式では，データを複数データセンタに振り分けて登録するが，この際，あらかじめ秘密情報として設定した「**振り分け確率**」に従って振り分けを行う．振り分け確率は例えば**表 5** のように設定され，それぞれの確率は「当該キーワードが設定されたデータの登録先として $\mathcal{C}_i$ を選択する確率」を表している．ここで，キーワードごとに異なる振り分け確率を設定することが本質であり，これによって各データセンタにおけるタグの頻度が撹乱される．

表 5 振り分け確率（秘密情報）

キーワード	$\mathcal{C}_1$	$\mathcal{C}_2$
東京	0.362	0.638
神奈川	0.585	0.415
大阪	0.845	0.155
愛知	0.151	0.849
埼玉	0.645	0.355
⋮	⋮	⋮

なお，キーワード集合が事前に判明している場合は振り分け確率を表 5 のように決めておけば良いが，キーワード集合が事前に分からない場合であっても，例えば「キーワードの鍵付きハッシュ値から振り分け確率を導出する」のように決めておくことで，提案方式を同様に適用できる．

3.2 アルゴリズム

前述の振り分け確率を取り入れた，提案方式の処理手順を以下に示す．

準備

(1) $\mathcal{S}$ は，データ秘匿用の鍵，キーワード秘匿用の鍵を生成する．また，データ・キーワードの暗号化に用いる鍵を $\mathcal{R}$ に通知する．

(2) $\mathcal{S}$ は，秘密情報として振り分け確率を設定し，$\mathcal{R}$ に通知する．

登録

(3) $\mathcal{R}$ は，登録するデータ d に対し，登録キーワード w を設定する．簡単のため，ここでは各データに対し，単一の登録キーワードを設定するものとする．

(4) $\mathcal{R}$ は，登録キーワード w に関する振り分け確率に従い，登録先 $\mathcal{C}^*$ を決定する．

(5) $\mathcal{R}$ は，データ d およびキーワード w を暗号化し，得られた暗号化データ $E_1(d)$ およびタグ $E_2(w)$ を，(4) で決定した $\mathcal{C}^*$ に送信する．

(6) $\mathcal{C}^*$ は，$\mathcal{R}$ から受信した $E_1(d), E_2(w)$ を対応付けたうえで，暗号化 DB に登録する．

検索

(7) $\mathcal{S}$ は，検索キーワード w' に対応する検索要求 $Q(w')$ を生成し，**全ての** $\mathcal{C}_j$ に送信する．

(8) 各 $\mathcal{C}_j$ は，暗号化 DB の各レコードについて，タグ $E_2(w_i)$ が検索要求 $Q(w')$ とマッチするかを確認する．

(9) 各 $\mathcal{C}_j$ は，マッチしたタグに対応する暗号化データ $E_1(d_i)$ を $\mathcal{S}$ に送信する．

(10) $\mathcal{S}$ は，全ての $\mathcal{C}_j$ から受信した $E_1(d_i)$ を復号し，データ d_i を得る．

表 6　タグの頻度分布（括弧内は各 DB における比率）

タグ	$\mathcal{C}_1$	$\mathcal{C}_2$
E_{D}(東京)	3825（0.071）	6703（0.146）
E_{D}(神奈川)	4355（0.080）	3038（0.066）
E_{D}(大阪)	5790（0.107）	1092（0.024）
E_{D}(愛知)	845（0.016）	5065（0.111）
E_{D}(埼玉)	3698（0.068）	1986（0.043）
⋮	⋮	⋮
総数	54228	45772

3.3　効果：タグの頻度撹乱による頻度分析耐性向上

提案方式を用いると，各データセンタで観測されるタグの頻度が撹乱されることを実例で示す．

日本全国から一様に選んだ 10 万人分のデータを，都道府県名をキーワードとして登録する際，表 5 の確率に従って 2 個のデータセンタに振り分ける場合を考える．このとき，キーワードを東京とするデータ（全体の 10.6%程度）は，確率 0.362 で $\mathcal{C}_1$ に，確率 0.638 で $\mathcal{C}_2$ に登録されるため，全体の 3.8%程度が $\mathcal{C}_1$ に，全体の 6.8%程度が $\mathcal{C}_2$ に登録されることになる．他の都道府県についても同様に考えると，各データセンタに登録されるタグの頻度は例えば**表 6** のようになり，キーワードの頻度から大きく変化していることが分かる．したがって，振り分け確率を秘密情報として扱う限り，各データセンタにおけるタグの頻度からキーワードを推定することが困難となる（4 節で定量的に評価する）．

なお，提案方式ではキーワードごとに異なる振り分け確率を設定することが本質である．キーワードを考慮せず無作為に振り分けを行った場合，各データセンタにおけるタグの頻度分布は，キーワードの頻度分布からほぼ変化しないことになり，頻度分析対策としての効果が得られない．

一方，高速化を目的とした複数 DB の活用法として，キーワードごとに決められたデータセンタに登録（例えば北海道・東北地方のデータを $\mathcal{C}_1$ へ，関東地方のデータを $\mathcal{C}_2$ へ，など分散）し，検索時は必要なデータセンタのみにアクセスすることでデータセンタの計算量を削減する方式がある．この場合，各データセンタに登録されるタグの頻度分布が変化するので，一見すると頻度分析対策も兼ねるようにも思えるが，実際には各タグの頻度比がキーワードの頻度比から変化しないため，頻度分析対策としてはあまり効果がない（以上についても 4 節で評価する）．

3.4　効率

まず，データセンタ側から見た場合，提案方式を用いた場合の処理は，単一データセンタを利用する場合の処理と全く変わらない．また，全データセンタが扱うデータの総量（ストレージ・計算量・通信量）も変わらないため，オーバーヘッドを生じることなく秘匿検索を実現できる．

データ登録者，データ検索者から見た場合も，追加の処理は登録先の決定，検索結果の集約などごくわずかであり，追加の情報も振り分け確率（または鍵付きハッシュ用の鍵）だけであるため，効率面でのデメリットはほとんどないと考えて良い．

4.　安全性評価

本節では，既存方式と提案方式について，検索で判明する頻度を利用した頻度分析攻撃への耐性を定量的に評価した結果を記す．

4.1　評価対象

本論文では，データ量や通信量などのオーバーヘッドを生じない方式を評価対象とし，下記の 4 方式について評価を行った．方式 1〜3 が既存方式に相当し，方式 4 が提案方式である．

- 方式 1：単一 DB
  頻度分析攻撃への対策を何も行わない．すなわち，全てのデータを単一 DB に登録する方式．
- 方式 2：均一振り分け
  登録データを複数 DB に振り分けるが，キーワードによらず，登録先 DB を一様ランダムに決定する方式．
- 方式 3：確定的振り分け
  秘匿検索対象となる各キーワードに対し，単一の登録先 DB をあらかじめ決めておく方式．
- 方式 4：確率的振り分け（提案方式）
  3 節で述べた提案方式．すなわち，秘匿検索対象となる各キーワードに対し，各 DB への振り分け確率をあらかじめ決めておく方式．

4.2　評価指標

本節では，本論文で安全性評価に用いた指標と，評価の中で攻撃者が行う最尤推定について説明する．

4.2.1　評価指標：攻撃者によるキーワード推定の的中率

本論文では，下記の条件下で行ったキーワード推定の的中率を評価指標として利用する．

キーワード

- キーワードは有限集合とし，要素数を k とする．
- 各キーワードの頻度が決まっており，登録データのキーワードは，前記頻度に従って発生する．

攻撃者

- 各データセンタを，キーワード推定を行う攻撃者とする．データセンタ同士は結託しない．
- 各データセンタは，キーワードの頻度を知っている．ただし，保持する DB における各キーワードの出現数は分からない．
- 各データセンタは全キーワードに関する検索要求を受信しており，保持する DB における全タグの頻度（出現数）を観測できる．

表 7　タグの頻度と攻撃者による推定の例

	$\mathcal{C}_1$			$\mathcal{C}_2$		
タグ	頻度	推定	結果	頻度	推定	結果
E_{D}(東京)	17	東京	的中	17	東京	的中
E_{D}(神奈川)	10	神奈川	的中	7	愛知	×
E_{D}(大阪)	7	愛知	×	11	大阪	的中
E_{D}(愛知)	9	大阪	×	12	神奈川	×
E_{D}(埼玉)	5	埼玉	的中	5	埼玉	的中
総数	48			52		

推定方法

- 各データセンタは，k 種類のキーワードと，k 種類[*1]のタグの対応付けを，最尤推定によって推定する．

なお，的中率は「キーワード推定が的中したレコード数の，全データセンタにおける総和」を「全データセンタに登録したレコード数」で割ったもので定義する．例えば，タグの頻度と攻撃者による推定が**表 7** のようになった場合，$\mathcal{C}_1$ で 32 レコード，$\mathcal{C}_2$ で 33 レコードが的中しているので，的中率は $(32+33)/100 = 0.65$ で 65%となる．

4.2.2　攻撃者による最尤推定

本節では，前節で述べた最尤推定について説明する．

まず方式 1 に対する推定（キーワードとタグの対応付けの推定）を考えると，最尤推定は「キーワードの頻度とタグの頻度をそれぞれ降順に並べ，頻度の高い順にキーワードとタグを対応付けたもの」となる．都道府県の例では，タグの頻度が高いものから順に東京，神奈川，大阪，と推定したものが最尤推定である．以下にその理由を示す．なお，ここでは簡単のため，各キーワードの頻度，各タグの頻度はそれぞれ異なっているものとして説明する．

観測したタグを頻度の降順に並べたものをタグ 1〜タグ k とし，タグ i の出現数を t_i とする（$t_i > t_{i+1}$）．また，攻撃者によるタグ i の推定をキーワード i とし，キーワード i の頻度（確率）を p_i とする．この推定（タグ i とキーワード i がそれぞれ対応する）が完全に正しいとした場合，キーワードの頻度分布から，各タグ i（=キーワード i）が t_i 個観測される確率（尤度）は次の値となる．

$$\frac{(\sum t_i)!}{\prod (t_i!)} \prod_{i=1}^{k} {p_i}^{t_i}$$

ここで，推定から定まる $\{p_i\}$ の中に，$p_i < p_{i+1}$ なる 2 要素が含まれていた場合，p_i, p_{i+1} の順番を入れ替えた方が尤度が高い．なぜなら，$t_i > t_{i+1}, p_i < p_{i+1}$ のとき，

$$\frac{{p_{i+1}}^{t_i} {p_i}^{t_{i+1}}}{{p_i}^{t_i} {p_{i+1}}^{t_{i+1}}} = \left(\frac{p_{i+1}}{p_i}\right)^{t_i - t_{i+1}} > 1.$$

つまり，最尤推定（キーワードが有限集合なので存在は自明）で定まる $\{p_i\}$ は，全ての i について $p_i > p_{i+1}$ を満たす必要がある．したがって，タグを頻度の降順に並べたとき，キーワードも頻度の降順に並べて対応付けたものが最尤推定となることが示された．

方式 2，4 に対しても，「頻度の高いタグは頻度の高いキーワードから生成された可能性が高い」という性質は変わらないため，方式 1 と同様に最尤推定を行うことができる．一方，方式 3 では，各データセンタは k 種類あるキーワードのうち，一部に対応付いた k' 種類（$k' < k$）のタグのみを観測する．したがって方式 1 に対する推定をそのままでは適用できないが，k 種類のキーワードから k' 種類を選ぶ全ての組み合わせについて，当該 k' 種類のキーワードだけが発生すると仮定した場合の最尤推定を行い，全組み合わせの中で最大尤度となるものを選択することで，最尤推定を行うことができる．

4.3　評価方法

的中率は，下記の手順で計算機シミュレーションを実施して評価した．なお，n は DB に登録する総レコード数，m は DB 数を表す．

(1) 各キーワードに対し，振り分け確率を決定．
(2) キーワードの頻度分布に従い，n 個のキーワードをランダムに決定．
(3) 各キーワードの振り分け確率に従い，n 個のキーワードの登録先をランダムに決定．
(4) DB ごとに，当該 DB におけるタグの頻度のみを観測した攻撃者による，キーワードの最尤推定を実施．
(5) DB ごとのキーワード的中数を合計し，n で割ったものを（3）の決定に対する的中率とする．
(6) （3）に戻ってランダム決定を 50 回繰り返し，的中率の平均値を（2）の決定に対する的中率とする．
(7) （2）に戻ってランダム決定を 50 回繰り返し，的中率の平均値を（1）の決定に対する的中率とする．
(8) （1）に戻ってランダム決定を 50 回繰り返し，的中率の平均値を評価対象方式に対する的中率とする．

手順（1）に関する補足を以下に記す．

- 方式 1：単一 DB のため振り分け確率は設定しない．
- 方式 2：キーワードによらず，各 DB への振り分け確率を $1/m$ とする．
- 方式 3：DB のうち一つをランダムに選択して振り分け確率を 1 とし，他 DB への振り分け確率を 0 とする．
- 方式 4：$[0, 1)$ の一様乱数を m 個生成し，合計が 1 となるよう正規化して各 DB への振り分け確率とする．

4.4　評価結果

4.3 節までで述べた条件のもとで，DB に登録するレコード数，出現するキーワードの種類数，DB 数を変化させ，攻撃者による推定の的中率を評価した結果を記す．

4.4.1　頻度分布の実例に対する各方式の安全性比較

まずは現実に即した例として，DB の各レコードに日本居住者の情報が登録されており，このうち都道府県名を秘

[*1] 出現数 0 で観測されなかったタグも仮想的に含める．

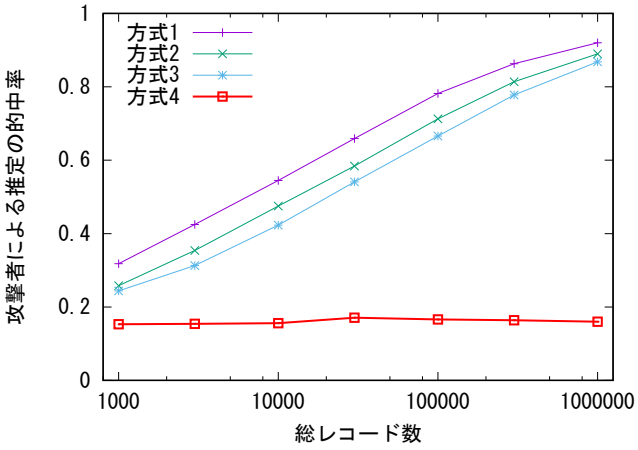

図 1　都道府県名を秘匿検索対象とした場合の安全性比較

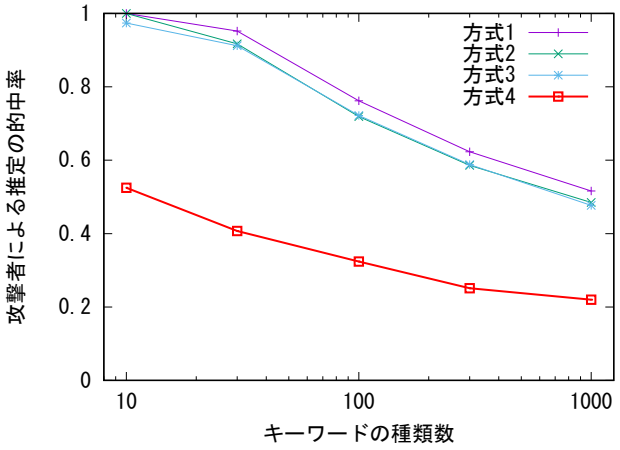

図 2　異なる頻度分布での的中率（頻度が順位に反比例する場合）

匿検索対象とする場合を想定し，各方式の安全性（攻撃者による都道府県推定の的中率）を比較した．なお，方式 2〜4 における DB 数は 2 個とし，都道府県の頻度分布は平成 27 年国勢調査の結果 [9] から算出した（表 4）．

評価結果は図 **1** のとおり．要点を以下に記す．

- 無対策（方式 1）の場合，例えば 10 万レコードのうち 78%のキーワードが的中してしまう．頻度分析攻撃は現実的な脅威であると言える．
- 単に複数 DB への分散を行うだけ（方式 2，3）では，頻度分析対策としての効果は限定的．
- 提案方式（方式 4）は，レコード数によらず，頻度分析対策として安定した効果を持つ（的中率 15〜17% [*2]）．

4.4.2　キーワードの種類数を変化させた場合

次に，頻度分布が異なる場合の的中率を見るために，キーワードの種類数を変化させて評価を行った．頻度分布については，i 番目のキーワードの頻度が i に反比例する場合（例えば 10 キーワードの場合，頻度の高い方から $34.1\%, 17.1\%, 11.4\%, 8.5\%, \ldots$），$i$ 番目のキーワードの頻度が i に比例する場合（例えば 10 キーワードの場合，頻度の高い方から $18.2\%, 16.4\%, 14.5\%, 12.7\%, \ldots$）の 2 タイプについて評価した．なお，総レコード数は 10 万件，方式 2〜4 における DB 数は 2 個とした．

評価結果はそれぞれ図 **2**，図 **3** のとおり．様々な頻度分布において，提案方式が安定した効果を持つことが分かる．

4.4.3　DB 数を変化させた場合

提案方式における最適な DB 数を論じるために，総レコード数を 10 万件に固定したうえで，DB 数を変化させて評価を行った（方式 2，3 も併せて評価）．なお，頻度分布については 4.4.1 節同様，都道府県のものを利用した．

評価結果は図 **4** のとおり．提案方式の効果は DB 数にほとんど依存せず，したがって 2 個の DB で十分な効果が得られることが分かる．

[*2] 東京の人口比率が 10.6%であるため，どんな対策をしても 10.6%の的中は可能である点に注意．

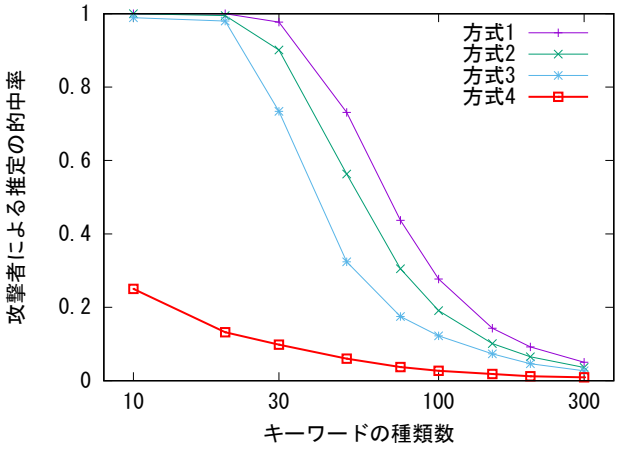

図 3　異なる頻度分布での的中率（順位ごとの頻度が等差の場合）

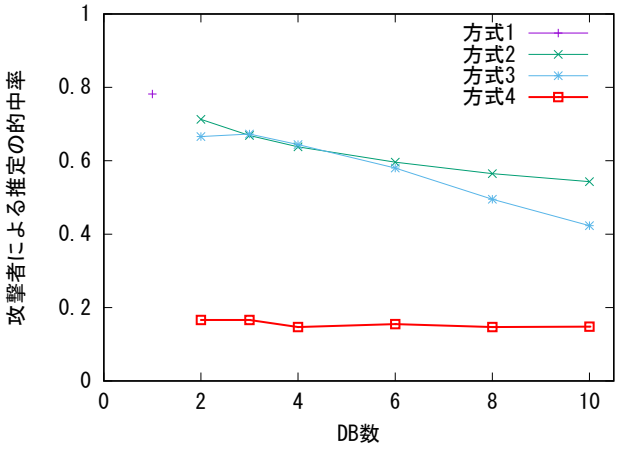

図 4　総レコード数を固定し，DB 数を変化させた場合の的中率

5.　まとめ

本論文では，既存の秘匿検索システムが，検索で判明する頻度を利用した頻度分析攻撃に弱いという課題に対し，頻度撹乱の手段として複数 DB を活用し，キーワードごとに設定した振り分け確率に従って登録先 DB を選択するこ

とで，効率面でのオーバーヘッドを生じることなく，頻度分析対策を実現する方式を提案した．また，頻度分析攻撃への耐性を定量評価するための指標を，最尤推定を行う攻撃者によるキーワード的中率として定義し，提案方式が頻度分析対策として安定した効果を持つことを述べた．提案方式の本質は登録先のランダム選択という簡潔な処理であるため，既存の多くの秘匿検索方式と組み合わせて使えるという利点を持つ．

今後の課題としては，本論文で定義した評価指標を，オーバーヘッドを生じる他の方式にも適用して比較することや，システムを実際に構築して性能評価を行うことなどが挙げられる．

参考文献

[1] Song, D.X., Wagner, D. and Perrig, A.: Practical Techniques for Searches on Encrypted Data, *Proc. IEEE Symposium on Security and Privacy*, IEEE Computer Society, pp.44–55 (2000).

[2] Boneh, D., Di Crescenzo, G., Ostrovsky, R. and Persiano, G.: Public Key Encryption with Keyword Search, *Proc. Advances in Cryptology - EUROCRYPT 2004*, LNCS 3027, pp.506–522 (2004).

[3] Curtmola, R., Garay, J., Kamara, S. and Ostrovsky, R.: Searchable Symmetric Encryption: Improved Definitions and Efficient Constructions, *Proc. 13th ACM Conference on Computer and Communications Security*, pp.79–88 (2006).

[4] Bellare, M., Boldyreva, A. and O'Neill, A.: Deterministic and Efficiently Searchable Encryption, *Proc. Advances in Cryptology - CRYPTO 2007*, LNCS 4622, pp.535–552 (2007).

[5] Kamara, S., Papamanthou, C. and Roeder, T.: Dynamic Searchable Symmetric Encryption, *Proc. 2012 ACM Conference on Computer and Communications Security*, pp.965–976 (2012).

[6] 清本晋作，田中俊昭，三宅優，三田村好矩，乗越雅光：全文検索可能な暗号化 DB の実装・評価，コンピュータセキュリティシンポジウム 2003 論文集，Vol.2003, No.15, pp.301–306 (2003).

[7] 伊藤隆，服部充洋，松田規，坂井祐介，太田和夫：頻度分析耐性を持つ高速秘匿検索方式，電子情報通信学会技術研究報告，Vol.110, No.443, pp.1–6 (2011).

[8] 山岡裕司，牛田芽生恵，伊藤孝一：頻度分析を k -匿名性で緩和する検索可能共通鍵暗号，コンピュータセキュリティシンポジウム 2015 論文集，Vol.2015, No.3, pp.568–575 (2015).

[9] 総務省統計局：平成 27 年国勢調査 人口等基本集計結果（オンライン），入手先 ⟨http://www.stat.go.jp/data/kokusei/2015/kekka.htm⟩（参照 2017-05-10）.