社団法人 情報処理学会 研究報告
IPSJ SIG Technical Reports

2005－ＣＳＥＣ－30 (53)
2005／7／22

検索語の秘匿と検索結果の一貫性検証を可能とするデータ検索プロトコルにおける通信量の削減

中山 敏† 吉田 真紀† 岡村 真吾† 藤原 晶† 藤原 融†

† 大阪大学大学院情報科学研究科
〒 565–0871 大阪府吹田市山田丘 1–5
E-mail: †{s-nakaym@ist, maki-yos@ist, s-okamur@ist, a-fujiwr@iff.ics.es, fujiwara@ist}.osaka-u.ac.jp

あらまし データ検索では，検索語をもつユーザが検索対象であるデータベースをもつサーバに対して問い合わせを行い，それに対する返答から検索結果としてデータベースの情報を得る．データ検索に対する安全性要件として，検索語の秘匿，データベースの秘匿，検索結果の一貫性検証がある．著者等はこれまでに，三つの安全性要件を満たしたデータ検索プロトコルを提案した．サーバは Merkle 木を用いて一貫性検証を可能とする値（コミット値と証明値）を生成する．ユーザは Oblivious Transfer (OT) を用いて検索結果と証明値をサーバから得る．データ検索では通信量が少ないことが望まれるため，サーバはユーザから受け取った検索結果に対する問い合わせを変換して，証明値に対する問い合わせとしても利用している．しかし，変換後の問い合わせに対する返答の通信量がデータベースの大きさに比例してしまい効率が悪い．そこで，本稿では変換後の問い合わせに対する返答の通信量がデータベースの大きさに比例しない OT を提案し，それを用いることによって通信効率の良いデータ検索プロトコルを実現した．

キーワード データ検索，検索語の秘匿，データベースの秘匿，検索結果の一貫性検証，Merkle 木，Oblivious transfer，通信効率

Reducing Communication Complexity of the Private Data Retrieval Protocol with Consistent Results

Satoshi NAKAYAMA†, Maki YOSHIDA†, Shingo OKAMURA†, Akira FUJIWARA†, and Toru FUJIWARA†

† Graduate School of Information Science and Technology, Osaka University
1–5 Yamadaoka, Suita-shi, Osaka, 565–0871 Japan
E-mail: †{s-nakaym@ist, maki-yos@ist, s-okamur@ist, a-fujiwr@iff.ics.es, fujiwara@ist}.osaka-u.ac.jp

Abstract A data retrieval protocol between a database server, who has a database, and a user, who has an index, allows the user to obtain an item in the database. Security requirements for data retrieval are the privacy of a user, the privacy of the database server, and the consistency of an answer. We have proposed a data retrieval protocol which satisfies the three security requirements. The server uses Merkle tree to generate a commitment and a proof which enable the user to verify the consistency of an answer, and publishes the commitment. The user executes Oblivious transfer (OT) with the server to obtain not only the item but also the proof. To make the protocol efficient, the server transforms a query for the item into a query for the proof and uses it. However, the size of an answer for a query obtained by the transformation is linear in the size of the database. That is, the previous protocol is inefficient. In this paper, we propose an efficient OT such that the size of an answer for the query obtained by the transformation is not linear in the size of the database, and realize an efficient data retrieval protocol by using the proposed OT.

Key words Data Retrieval, Privacy of a User, Privacy of the Server, Consistency of an Answer, Merkle Tree, Oblivious Transfer, Efficiency

1. まえがき

近年，様々な情報が電子化され，サーバに蓄積・管理されるようになってきている．それに伴い，サーバに蓄積されたデータを利用する機会も増加してきている．サーバに蓄積されたデータの利用法の一つとして，データ検索 [1] が挙げられる．データ検索とは，検索対象であるデータベースをもつサーバと，検索語をもつユーザの間で実行される．ユーザは，検索語を用いてサーバに対して問い合わせを行い，サーバの返答から検索結果としてデータベースに関する情報を得る．様々な検索語，検索対象，検索結果に対するデータ検索が考えられる．例えば音楽配信であれば，検索語が曲名，検索対象が音楽データ群，検索結果が検索対象の音楽データの中で検索語を曲名とする音楽データとなるデータ検索が挙げられる．

これまでに，様々な種類のデータ検索が考えられ，安全性要件を満たすデータ検索プロトコルが設計されている [2]〜[7]．データ検索に対する安全性要件として，検索語の秘匿 [4], [5], [7]，データベースの秘匿 [2]〜[7]，検索結果の一貫性検証 [2], [3], [6], [7] の三つがある．検索語の秘匿とは，ユーザの検索語をサーバから秘匿できることである．検索語はユーザの嗜好やプライバシに関わることが考えられるため，この安全性要件を満たすことが要求される．なお，検索語が秘匿されることは検索結果も秘匿されることを意味する．これは，検索結果が分かれば検索語を絞り込むことができるため，検索語に関する何らかの情報を得ていると考えられるからである．データベースの秘匿とは，サーバがもつデータベースの情報を本来の検索結果から分かること以外はユーザから秘匿できることである．サーバが検索サービスを有料で提供しており，問い合わせごとに課金する場合など，データベースの内容がサーバにとって貴重な情報である場合，この安全性要件を満たすことが要求される．検索結果の一貫性検証とは，サーバが検索語に対する検索結果を変えていないこと，すなわち検索結果が一貫していることを証明できることである．具体的には，サーバが予めデータベースのコミット値を公開し，検索時にユーザが検索結果だけでなく一貫性検証のための値（証明値）を入手でき，検索結果が検索語とコミットされたデータベースに対応していることを検証できることである．

各安全性要件を満たす自明な方法を考えた場合，問い合わせ内容，返答内容，コミット値および証明値の大きさの上限は，それぞれデータベースの大きさとなる．例えば，検索語の秘匿と検索結果の一貫性を満たすのであれば，コミット値としてデータベースの内容を全て公開すれば良い．よって，通信効率に関する要件は，問い合わせ内容，返答内容，コミット値および証明値の大きさをそれぞれデータベースの大きさよりも小さくすることがある．

これまでに，三つの安全性要件を全て満たしたデータ検索プロトコルとして，Oblivious Keyword Search (OKS) [7] が提案されている．OKS では，問い合わせ内容と返答内容の大きさがデータベースの大きさに依存しないため，検索の際の通信効率は非常に良いといえる．しかし，コミット値がデータベースの各データを暗号化したものであり，データベースの大きさに比例した大きさとなるため，通信効率に関する要件を完全には満たしていない．一方，二つの安全性要件を満たしたデータ検索プロトコルには，通信効率に関する要件を満たしたものがある [2]〜[6]．その中で最も効率が良いものを挙げると，検索語の秘匿とデータベースの秘匿を満たしたものとして Oblivious Transfer (OT) [4] が挙げられる．また，データベースの秘匿と検索結果の一貫性検証を満たしたものとして Merkle 木 (MT) [10], [11] を利用した Undeniable Replies (UR) [2], [3] が挙げられる．

本研究では，三つの安全性要件を全て満たし，通信効率に関する要件も満たしたデータ検索プロトコルの設計を目的とする．データ検索のモデルとして，OT [4] や Symmetrically Private Information Retrieval [5] と同じモデルを対象とする．

三つの安全性要件と通信効率に関する要件を満たすデータ検索プロトコルの自明な構成法は，二つの安全性要件を満たす効率の良いデータ検索プロトコルである OT [4] と UR [2] を組み合わせることである．具体的には，サーバがコミット値と証明値を生成するために，UR [2] と同様 MT を利用し，ユーザが検索結果と証明値を得るために OT [4] を利用する．MT を利用することで，コミット値の大きさをデータベースの大きさに依存せず定数に，証明値の大きさをデータベースの大きさの対数にできる．著者らは [12] で自明な構成法より通信効率を向上させる設計方針を示した．それは，問い合わせの通信量を削減するために，サーバが検索結果を得るための問い合わせ内容を証明値を得るための問い合わせ内容に変換して利用するというものである．しかし，その設計方針の下で MT と OT [4] を用いて構成したデータ検索プロトコルでは，返答の通信量がデータベースの大きさに比例し，OKS とは逆に検索の際の通信効率が悪くなってしまった．

そこで本稿では，[12] で示した設計方針に適する OT を [4] の OT を拡張することで構成し，その OT を利用することで通信効率に関する要件も満たすデータ検索プロトコルを実現する．

本稿の構成を以下に示す．**2.** では本稿で対象とするデータ検索のモデルを示す．**3.** では提案するデータ検索プロトコルで利用する暗号技術である MT，Paillier 暗号，OT の定義を示す．**4.** では [12] で提案した通信効率を向上させる設計方針を示す．**5.** ではまず OT [4] を拡張した OT を提案し，その OT の問い合わせの変換法を示した上でデータ検索プロトコルを提案する．**6.** では提案プロトコルの安全性と効率を評価し，最後に **7.** で全体のまとめと今後の課題について述べる．

2. 対象とするデータ検索モデル

本稿で対象とするデータ検索モデルを示す．**2.1** ではデータ検索における参加者と処理を示し，**2.2** ではデータ検索に対する要件を示す．

2.1 参加者と処理

自然数 x に対して，$[x]$ は 1 から x までの自然数からなる集合を表すものとする．参加者は検索語 i ($i \in [N]$) をもつユーザと，検索対象データベース $D = \{d_i \mid i \in [N]\}$ をもつサーバ

の二者である．各データ d_i の大きさは全て等しく，N に依存しないとし，その大きさを l_d とする．データ検索における処理は，コミット値と証明値の生成，問い合わせ内容の生成，返答内容の生成，検索結果の入手，証明値の入手，検索結果の一貫性検証の六つからなる．各処理の内容を以下に示す．

コミット値と証明値の生成 サーバが実行する．データベース D のコミット値 $C(D)$ と，D の各データに対する証明値 $p(i,D)$ ($i \in [N]$) を生成する．生成されたコミット値はサーバによって予め公開される．この処理は，一度実行されるとデータベースの内容が更新されるまで実行されることはない．

問い合わせ内容の生成 ユーザが実行する．検索語 i に対する問い合わせ内容 $Q(i)$ を生成する．生成された $Q(i)$ はサーバに送信される．

返答内容の生成 サーバが実行する．問い合わせ内容 $Q(i)$ とデータベース D に対して，ユーザが検索結果 d_i と証明値 $p(i,D)$ を入手するための返答内容 $A(Q(i),D)$ を生成する．生成された $A(Q(i),D)$ はユーザに送信される．

検索結果の入手 ユーザが実行する．返答内容 $A(Q(i),D)$ から i に対する検索結果 d_i を得る．

証明値の入手 ユーザが実行する．返答内容 $A(Q(i),D)$ から d_i に対する証明値 $p(i,D)$ を得る．

検索結果の一貫性検証 ユーザが実行する．検索結果 d_i，証明値 $p(i,D)$，コミット値 $C(D)$ から検索語 i とコミットされた D に対応する検索結果は d_i しかないことを検証する．

2.2 データ検索に対する要件

本稿では，データ検索に対する要件として，安全性に関する三つの要件と通信効率に関する一つの要件を考える．安全性に関する要件は，次の三つである．

要件 1（検索語の秘匿） サーバは，検索語 i を求めることができない．

要件 2（データベースの秘匿） ユーザは，データベース D に関して本来の検索結果である d_i から分かること以外は分からない．よって，各データが互いに独立ならば，ユーザは $d_j \in D$ ($j \neq i$) を求めることができない．

要件 3（検索結果の一貫性検証） ユーザは，コミット値 $C(D)$ と証明値 $p(i,D)$ を用いて検索語 i とコミットされたデータベース D に対応する検索結果は d_i しか無いことを検証できる．

通信効率に関する要件は次の一つである．

要件 4（通信効率） 問い合わせ内容，返答内容，コミット値，証明値の大きさがそれぞれデータベースの大きさよりも小さい．すなわち，任意の i ($i \in [N]$) に対して以下が成り立つ．

$$|Q(i)|,\ |A(Q(i),D)|,\ |C(D)|,\ |p(i,D)| < N \cdot l_d$$

ここで，データ d に対して $|d|$ は d の大きさを表すとする．

3. 利用する暗号技術

提案するデータ検索プロトコルでは，暗号技術として Merkle 木および $(1,m)$-Oblivious Transfer（略して $(1,m)$-OT）を利用する．$(1,m)$-OT は [4] の $(1,m)$-OT を拡張したものを利用する．文献 [4] の $(1,m)$-OT および提案プロトコルで利用する $(1,m)$-OT では，Paillier 暗号を利用している．よって本章では，Merkle 木，Paillier 暗号および $(1,m)$-OT を紹介する．

3.1 Merkle 木

Merkle 木は，各節点に値を割り当てた完全 2 分木である．根節点の深さを 0，根節点を x_0 と表し，深さ h において左から j 番目の節点を $x_{h,j}$ と表す（ただし，$j \in [2^h]$）．また，節点 x の兄弟節点を $\bar{x}$ と表す．節点 x に割り当てる値を $V(x)$ と表し，値の割り当て方を示す．葉節点には任意の値を割り当てられる．各内部節点 x に割り当てる値は，その左の子節点と右の子節点に割り当てた値を連接した値に，衝突困難一方向性ハッシュ関数 H を適用した値である．すなわち，$V(x_{h,j}) = H(V(x_{h+1,2j-1}) \parallel V(x_{h+1,2j}))$ である．ここで，"$\parallel$" は連接を表す．

3.2 Paillier 暗号

Paillier 暗号は，公開鍵暗号方式の一つであり，加算に関して準同型性をもつ．すなわち，暗号化関数を E とするとき，任意の平文 M_1, M_2 について，$E(M_1) \cdot E(M_2) = E(M_1 + M_2)$ が成り立つ．

n を RSA 暗号で用いられる二素数積 ($n = p \cdot q$) とし，g を位数が n の倍数である $Z^*_{n^2}$ の要素とする．公開暗号化鍵は $k_e = (n, g)$ であり，秘密復号鍵は $k_d = \lambda(n) = \mathrm{lcm}(p-1, q-1)$ である．Paillier 暗号における暗号化と復号の各処理を示す．

暗号化 平文 $M \in Z_n$ に対して，暗号化に用いる乱数 $r \in Z^*_n$ を生成し次の式に従って暗号化を行う．

$$E_g(M,r) = g^M r^n \bmod n^2$$

以降では，乱数 r を用いて平文 M を暗号化した暗号文を公開暗号化鍵 g と乱数 r を省略して $E(M)$ と表す．

復号 暗号文 w に対して次の式に従って復号を行う．

$$D_g(w) = \frac{L(w^{\lambda(n)} \bmod n^2)}{L(g^{\lambda(n)} \bmod n^2)} \bmod n$$

ここで，$L(u) = (u-1)/n$ ($u \in Z^*_{n^2}$) である．以降では，暗号文 w を復号した結果を $D(w)$ と表す．

3.3 $(1,m)$-Oblivious Transfer

$(1,m)$-OT の定義と [4] で提案された構成法を示す．

3.3.1 定　　義

$(1,m)$-OT は，本稿で対象とするデータ検索のモデルに従い，安全性要件として要件 1（検索語の秘匿）と要件 2（データベースの秘匿）を満たすものである．すなわち，サーバは m 個のデータ $\{d_1, ..., d_m\}$ を入力とし，ユーザは検索語 i ($i \in [m]$) を入力とし i をサーバに知られることなく検索結果として i 番目のデータ d_i だけを得る．処理は問い合わせ内容の生成，返答内容の生成，検索結果の入手の三つからなり，返答内容は検索結果だけに関する返答内容となる．

3.3.2 文献 [4] で提案された構成法

文献 [4] で提案された，HyperCube 構造に基づく $(1,m)$-OT（略して $(1,m)$-HCOT と呼ぶ）を示す．$(1,m)$-HCOT は，通信効率に関わるパラメータとして $m = l^c$ を満たす l, c をもつ．サーバがもつ m 個のデータは c-HyperCube として扱われる．c-HyperCube における d_i の位置情報として次の式を満たす c

個の値の組 $\langle i_j \rangle_{j\in[c]}$ が用いられ，d_i を表すために $d(i_1,...,i_c)$ が用いられる．

$$i = 1 + \sum_{j=1}^{c} (i_j - 1) \cdot l^{c-j} \quad (i_j \in [l])$$

$(1,m)$-HCOT における各処理を以下に示す．なお，処理の詳細は $c=2$ である場合について示す．$c=2$ は，通信効率に関する要件を満たしながら返答内容の大きさを最小とするパラメータである．

問い合わせ内容の生成 検索語 i，パラメータ l,c を入力とし，問い合わせ内容 HCOT-$Q(i)_{\langle l,c\rangle}$ を生成する．HCOT-$Q(i)_{\langle l,c\rangle}$ は，各組 l 個の暗号文 c 組からなる．$c=2$ の場合の処理の詳細は以下の通りである．

（1） Paillier 暗号の公開暗号化鍵と秘密復号鍵の組を生成する．以降では，暗号化および復号処理は全て Paillier 暗号を利用する．

（2） i に対して $\langle i_1, i_2\rangle$ を生成し，各 $i_j\,(j\in[2])$ に対して l 個の暗号文の組 $\langle E(I(t,i_j))\rangle_{t\in[l]}$ を生成する．ここで，I は二つの引数が一致するときに 1 を出力しそれ以外のときは 0 を出力する指標関数を表す．よって，l 個の暗号文のうち i_j 番目は 1 を，残りの $l-1$ 個は 0 をそれぞれ暗号化したものとなる．

（3） 生成した各 l 個の暗号文 2 組 $\langle E(I(t,i_j))\rangle_{t\in[l],j\in[2]}$ を問い合わせ内容 HCOT-$Q(i)_{\langle l,2\rangle}$ とする．以降では，$E(I(t,i_j))$ を $HQ(i)_{t,j}$ と表す．

返答内容の生成 問い合わせ内容 HCOT-$Q(i)_{\langle l,c\rangle}$，$m$ 個のデータ $\{d_1,...,d_m\}$，パラメータ l,c を入力とし，返答内容 HCOT-$A(Q(i)_{\langle l,c\rangle},D)_{\langle l,c\rangle}$ を生成する．HCOT-$A(Q(i)_{\langle l,c\rangle},D)_{\langle l,c\rangle}$ は，2^{c-1} 個の暗号文からなる．$c=2$ の場合の処理の詳細は以下の通りである．

（1） m 個のデータを l 個ずつ l 組に組み分けし，$Q(i)_{\langle l,2\rangle}$ を用いて各組の中で i_2 番目のデータを暗号化した暗号文を生成する．具体的には，各組に対して次の計算を行う．

$$W_j = \prod_{t\in[l]} HQ(i)_{t,2}^{d(j,t)} \bmod n^2 \quad (j\in[l])$$

Paillier 暗号の性質と暗号文の内容から，$W_j\,(j\in[l])$ は次の式を満たす．

$$W_j = E(d(j,i_2))$$

書き換えると，$d(j,i_2) = D(W_j)$ である．

（2） (1) で生成した l 個の暗号文 $\langle W_j\rangle_{j\in[l]}$ に対し，以下を満たす $\langle W1_j, W2_j\rangle$ $(W1_j, W2_j \in Z_n, j\in[l])$ を計算する．

$$W_j = W1_j \cdot n + W2_j$$

これは，暗号文と平文の値域が異なる Paillier 暗号において，暗号文と平文の間での演算を可能とするためである．

（3） (2) で生成した各 l 個の暗号文 $W1_j$ および $W2_j$ に対し，それぞれ i_1 番目の値を暗号化した暗号文を生成する．具体的には，$W1_j$ および $W2_j$ に対し次の計算を行う．

$$W\theta = \prod_{t\in[l]} HQ(i)_{t,1}^{W\theta_t} \bmod n^2 \quad (\theta\in[2])$$

$W1, W2$ は，m 個のデータの中の $(i_1-1)\cdot l + i_2$ 番目のデータを暗号化した暗号文を二つに分割後，更に暗号化した暗号文となる．すなわち，次の式を満たす．

$$d(i_1,i_2) = D(D(W1)\cdot n + D(W2))$$

$\langle W1, W2\rangle$ を返答内容 HCOT-$A(Q(i)_{\langle l,2\rangle},D)_{\langle l,2\rangle}$ とする．

検索結果の入手 返答内容 HCOT-$A(Q(i)_{\langle l,c\rangle},D)_{\langle l,c\rangle}$，パラメータ l,c，秘密復号鍵を入力とし，検索語 i に対応したデータ d_i を計算する．具体的には，返答内容である 2^{c-1} 個の暗号文を，生成された順番と逆の順番で復号することで，検索結果 $d_i = d(i_1,...,i_c)$ を得ることができる．$c=2$ の場合には，$D(D(W1)\cdot n + D(W2))$ を計算することで $d_i = d(i_1,i_2)$ を得ることができる．

$(1,m)$-HCOT を 1 回実行することでユーザは大きさが $\lceil \log n\rceil$ のデータを入手できる．その際の通信量を示す．問い合わせ内容の大きさは $(c\cdot l + \log m)\cdot\lceil 2\log n\rceil$ であり，返答内容の大きさは $(2^{c-1}+4\log m)\cdot\lceil 2\log n\rceil$ である．問い合わせ内容に含まれる $\log m\cdot\lceil 2\log n\rceil$ および返答内容に含まれる $4\log m\cdot\lceil 2\log n\rceil$ は，参加者がプロトコルに従わない malicious player であっても安全性要件を満たすために必要な暗号文の大きさである．ただし，簡単のため，この処理は $c=2$ の場合の説明に含めていない．

4. 設計方針

本稿で提案するデータ検索プロトコルの設計方針として，各処理をどのようにして実現するかを示す．なお，簡単のためデータベース内のデータ数 N を 2 のべき乗とする．

4.1 コミット値と証明値の生成および検索結果の一貫性検証

コミット値と証明値の生成および検索結果の一貫性検証に Merkle 木を利用する．利用法は [2] と同じであり，左から i 番目の葉節点には d_i のハッシュ値 $H(d_i)$ を割り当てる．すなわち，$V(x_{\log N,i}) = H(d_i)$ とする．これにより，Merkle 木の定義から内部節点に割り当てられる値も決定する．そして，根節点に割り当てた値をコミット値とする．すなわち，次の式が成り立つ．

$$C(D) = V(x_0)$$

ハッシュ関数の衝突困難性より，根節点の値を変更することなく葉節点の値を変更できないため，根節点は全ての葉節点の値に対するコミット値の役割を果たす．

節点 x および x の先祖全体からなる集合を $TA(x)$ と表す．節点 x およびその先祖それぞれの兄弟節点全体からなる集合を $TB(x)$ と表す．すなわち，$TB(x) = \{\overline{x'} \mid x'\in TA(x)\}$ となる．証明値 $p(i,D)$ は，検索語 i に対応づけられた葉節点とその先祖それぞれの兄弟節点に割り当てられた値全体からなる集合となる．すなわち，検索語 i と対応づけられた葉節点が $x_{\log N,i}$ となるため，次の式が成り立つ．

$$p(i,D) = \{V(x') \mid x'\in TB(x_{\log N,i})\}$$

証明値 $p(i,D)$ は，各深さ h において 1 つのハッシュ値を要素としてもつことになる．$p(i,D)$ の深さ h における要

素を $p(i,D)_h$ と表す．図 1 に例を示す．データベース内の

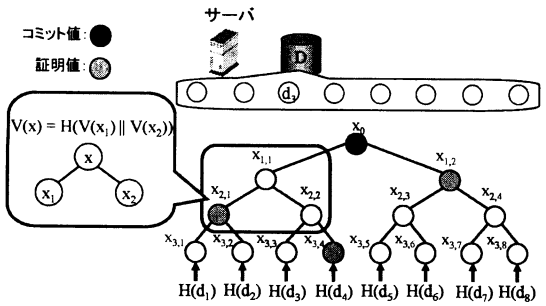

図 1　Merlke 木 ($N = 8$)

データ数 $N = 8$ および $i = 3$ のとき，証明値は内部節点 $x_{1,2}, x_{2,1}, x_{3,4}$ に割り当てられた値からなる．すなわち，$p(3,D) = \{V(x_{1,2}), V(x_{2,1}), V(x_{3,4})\}$ であり，$p(3,D)_1 = V(x_{1,2})$，$p(3,D)_2 = V(x_{2,1})$，$p(3,D)_3 = V(x_{3,4})$ である．

ユーザは，検索結果 d_i と証明値 $p(i,D)$ が入手できれば根節点に割り当てられた値を計算できる．よって，これが公開されているコミット値と等しいかを調べることで検索結果の一貫性を検証できる．

4.2　問い合わせ内容と返答内容の生成および検索結果と証明値の入手

ユーザは検索結果と証明値を得るために，サーバと OT を実行する．自明な OT の実行方法として，以下に示す二つの方法が挙げられる．なお，説明を簡単にするため，データベースの各データと MT におけるハッシュ値の大きさはそれぞれ OT の実行 1 回で送信できる大きさであるとする．

自明な方法 1　ユーザは検索結果 d_i を得るためにサーバとの間で $(1,N)$-HCOT を実行する．ユーザは検索語 i を入力とし，サーバはデータベース $D = \{d_1, ..., d_N\}$ を入力とする．これにより，ユーザは検索結果 d_i を得ることができる．次に，ユーザは証明値 $p(i,D)$ の各要素 $p(i,D)_h$ $(h \in [\log N])$ を得るために，サーバとの間で Merkle 木の各深さ h において $(1,2^h)$-HCOT を実行する．具体的には，ユーザは $p(i,D)_h$ が左から j 番目に位置するならば，すなわち $V(x_{h,j}) = p(i,D)_h$ ならば j を入力とし，サーバは 2^h 個のハッシュ値を入力とする．すなわち $D_h = \{d_{h,t} = V(x_{h,t}) \mid t \in [2^h]\}$ とする．これにより，ユーザは証明値 $p(i,D)$ を得ることができる．図 2 に例を示す．データベース内のデータ数 $N = 8$ および $i = 3$ のとき，ユーザはまず検索結果 d_3 を得るためにサーバとの間で $(1,8)$-HCOT を実行する．その後，証明値の要素 $V(x_{1,2})$，$V(x_{2,1})$，$V(x_{3,4})$ を得るためにそれぞれ $(1,2)$-HCOT，$(1,4)$-HCOT，$(1,8)$-HCOT を実行する．

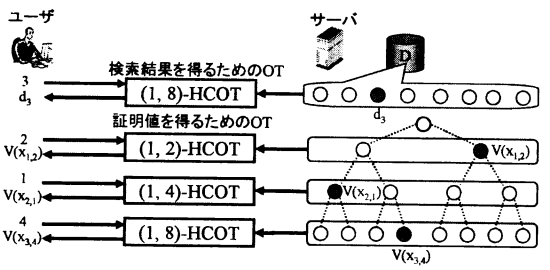

図 2　自明な方法 1 ($N = 8$)

自明な方法 2　サーバはデータベースの各データ d_i に対応する証明値 $p(i,D)$ を連接したデータ $d_i \parallel p(i,D)$ を生成する．ユーザは検索語 i を入力とし，サーバは $\{d_i \parallel p(i,D) \mid i \in [N]\}$ を入力として $(1,N)$-HCOT を実行する．ただし，$d_i \parallel p(i,D)$ の大きさが HCOT の実行 1 回で送信できるデータの大きさの $\log N + 1$ 倍となるため，サーバは $d_i \parallel p(i,D)$ を $\log N + 1$ 個に分割し，それぞれを入力として $(1,N)$-HCOT を実行する．その際，ユーザの入力 i は各 $(1,N)$-HCOT において同じであるため，ユーザは $(1,N)$-HCOT の実行 1 回分の問い合わせ内容を送信し，サーバはこれを再利用する．図 3 に例を示す．データベース内のデータ数 $N = 8$ および $i = 3$ のとき，ユーザは検索結果 d_3 を得るためにサーバとの間で $(1,8)$-HCOT を実行する．サーバはこの $(1,8)$-HCOT の問い合わせ内容と，各証明値 $p(i,D)$ の各分割結果を用いて，$(1,8)$-HCOT における返答内容を生成する．よって，ユーザは問い合わせ内容を $(1,8)$-HCOT の実行 1 回分送れば良い．

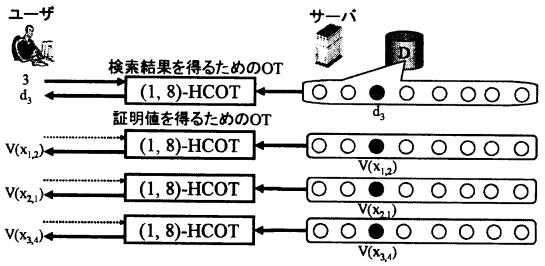

図 3　自明な方法 2 ($N = 8$)

自明な方法 1 と 2 には効率に関してそれぞれ長所と短所がある．それらを表 1 に示す．方法 1 は，問い合わせの通信量やユーザの問い合わせ内容生成処理が $(1,N)$-HCOT の実行 $(\log N + 1)$ 回分になってしまうが，返答の通信量やサーバの計算量は $(1,N)$-HCOT を繰り返す方法 2 に比べて小さく済む．方法 2 は，返答の通信量やサーバの計算量は方法 1 に比べて大きくなってしまうが，問い合わせの通信量やユーザの問い合わせ内容生成処理は $(1,N)$-HCOT の実行 1 回分で済む．

表 1　効率の比較

	通信量		計算量	
	問い合わせ	返答	ユーザ	サーバ
自明な方法 1	×	○	×	○
自明な方法 2	○	×	○	×

アイデア　本稿では，方法 1 の長所を残しながらその短所を解決したプロトコルを設計する．提案するプロトコルでは，方法 1 と同様に Merkle 木の各深さ h に対して $(1,2^h)$-OT を実行する．問い合わせの通信量を方法 2 と同様に $(1,N)$-HCOT

の実行1回分に削減するために，検索結果 d_i を得るための問い合わせ内容を証明値 $p(i,D)$ を得るための問い合わせ内容にサーバが変換できるようにする．変換を実現するために，まず，Merkle 木の各節点に割り当てた値をその節点の兄弟節点に割り当てた値と入れ替えた完全2分木 $\mathcal{T}$ を考える．この2分木 $\mathcal{T}$ において，検索語 i に対応する証明値の $\log N$ 個の要素は，葉節点 $x_{\log N,i}$ および $x_{\log N,i}$ の先祖節点の内，根節点以外の節点からなる．すなわち，$\{V(x) \mid x \in TA(x_{\log N,i}) \setminus x_0\}$ となる．よって，問い合わせ内容の変換は葉節点 $x_{\log N,i}$ に対する問い合わせ内容をその先祖節点 $x \in TA(x_{\log N,i}) \setminus x_0$ に対する問い合わせ内容に変換することに帰着できる．

変換を実現するために利用する OT を **5.1** で提案し，変換法を **5.2** で提案する．

5. 提案するデータ検索プロトコル

まず **5.1** で提案するデータ検索プロトコルで利用する $(1,m)$-OT を提案する．次に，**5.2** で問い合わせ内容の変換方法を示す．そして，**5.3** で提案するデータ検索プロトコルにおける処理を示す．なお，簡単のためデータベース内のデータ数 N を2のべき乗とする．

5.1 提案する $(1,m)$-OT

提案プロトコルで利用する OT を新たに提案する．この OT は $(1,m)$-HCOT を拡張したものであり，HyperCube 構造の代わりに木構造に基づく．以降ではこの OT を，$(1,m)$-TOT と表す．

$(1,m)$-TOT は，通信効率に関わるパラメータとして c と $m = l_1 \cdot l_2 \cdot \cdots \cdot l_c$ を満たす $l_1,...,l_c$ をもつ．ただし，$l_t \geq 2$ ($t \in [c]$) とする．サーバがもつ m 個のデータを，深さが c で各深さ h の内部節点が l_{h+1} 個の子節点をもつ木（以降では $(l_1,...,l_c)$-木と呼ぶ）の葉節点として扱う．$(l_1,...,l_c)$-木における d_i の位置情報として次の式を満たす c 個の値の組 $\langle i_j \rangle_{j\in[c]}$ を用い，d_i を表すために $d(i_1,...,i_c)$ を用いる．

$$i = \sum_{j=1}^{c-1}\Big((i_j - 1)\cdot \prod_{j+1\leq t\leq c} l_t\Big) + i_c \quad (i_j \in [l_j])$$

$(1,m)$-TOT における各処理を以下に示す．なお，処理の詳細は $c=2$ である場合について示す．$c=2$ は，通信効率に関する要件を満たしながら返答内容の大きさを最小とするパラメータである．

問い合わせ内容の生成 検索語 i，パラメータ $l_1,...,l_c,c$ を入力とし，問い合わせ内容 TOT-$Q(i)_{\langle l_1,...,l_c,c\rangle}$ を生成する．TOT-$Q(i)_{\langle l_1,...,l_c,c\rangle}$ は，暗号文 c 組からなり，j 番目の組は l_j 個の暗号文からなる（合計 $\sum_{j=1}^{c} l_j$ 個）．$c=2$ の場合の処理の詳細は以下の通りである．

（1） Paillier 暗号の公開暗号化鍵と秘密復号鍵の組を生成する．以降では，暗号化および復号処理は全て Paillier 暗号を利用する．

（2） i に対して $\langle i_1,i_2\rangle$ を生成し，各 i_j ($j \in [2]$) に対して l_j 個の暗号文の組 $\langle E(I(t,i_j))\rangle_{t\in[l_j]}$ を生成する．l_j 個の暗号文のうち i_j 番目は1を，残りの $l_j - 1$ 個は0をそれぞれ暗号化したものとなる．

（3） 生成した各 l_1+l_2 個の暗号文 $\langle E(I(t,i_j))\rangle_{t\in[l_j],j\in[2]}$ を問い合わせ内容 TOT-$Q(i)_{\langle l_1,l_2,2\rangle}$ とする．以降では，$E(I(i_j,t))$ を $TQ(i)_{t,j}$ と表す．

返答内容の生成 問い合わせ内容 TOT-$Q(i)_{\langle l_1,l_2,2\rangle}$，$m$ 個のデータ $\{d_1,...,d_m\}$，パラメータ $l_1,...,l_c,c$ を入力とし，返答内容 TOT-$A(Q(i)_{\langle l_1,...,l_c,c\rangle},D)_{\langle l_1,...,l_c,c\rangle}$ を生成する．返答内容は，2^{c-1} 個の暗号文からなる．$c=2$ の場合の処理の詳細は以下の通りである．

（1） m 個のデータを l_2 個ずつ l_1 組に組み分けし，$Q(i)_{\langle l_1,l_2,2\rangle}$ を用いて各組の中で i_2 番目のデータを暗号化した暗号文を生成する．具体的には，各組に対して次の計算を行う．

$$W_j = \prod_{t\in[l_2]} TQ(i)_{t,2}^{d(j,t)} \bmod n^2 \quad (j \in [l_1])$$

Paillier 暗号の性質と暗号文の内容から，W_j ($j \in [l_1]$) は次の式を満たす．

$$W_j = E(d(j,i_2)) \quad (j \in [l_1])$$

書き換えると，$d(j,i_2) = D(W_j)$ である．

（2） (1) で生成した l_1 個の暗号文 $\langle W_j\rangle_{j\in[l_1]}$ に対し，以下を満たす $\langle W1_j, W2_j\rangle$ $(W1_j, W2_j \in Z_n, j \in [l_1])$ を計算する．

$$W_j = W1_j \cdot n + W2_j$$

（3） (2) で生成した各 l_1 個の暗号文 $W1_j$ および $W2_j$ に対し，それぞれ i_1 番目の値を暗号化した暗号文を生成する．具体的には，$W1_j$ および $W2_j$ に対して次の計算を行う．

$$W\theta = \prod_{t\in[l_1]} TQ(i)_{t,1}^{W\theta_t} \bmod n^2 \quad (\theta \in [2])$$

$W1, W2$ は，m 個のデータの中の $(i_1-1)\cdot l_2 + i_2$ 番目のデータを暗号化した暗号文を二つに分割後，更に暗号化した暗号文となる．すなわち，次の式を満たす．

$$d(i_1,i_2) = D(D(W1)\cdot n + D(W2))$$

$\langle W1, W2\rangle$ を返答内容 TOT-$A(Q(i)_{\langle l_1,l_2,2\rangle},D)_{\langle l_1,l_2,2\rangle}$ とする．

検索結果の入手 返答内容 TOT-$A(Q(i)_{\langle l_1,...,l_c,c\rangle},D)_{\langle l_1,...,l_c,c\rangle}$，パラメータ $l_1,...,l_c,c$，秘密復号鍵を入力とし，検索語 i に対応したデータ d_i を計算する．具体的には，返答内容である 2^{c-1} 個の暗号文を，生成された順番と逆の順番で復号することで，検索結果 $d_i = d(i_1,...,i_c)$ を得ることができる．$c=2$ の場合には，$D(D(W1)\cdot n + D(W2))$ を計算することで $d_i = d(i_1,i_2)$ を得ることができる．

$(1,m)$-TOT を1回実行することでユーザは大きさが $\lceil \log n\rceil$ のデータを入手できる．その際の通信量を示す．問い合わせ内容の大きさは $(\sum_{j=1}^{c} l_c + \log m)\cdot\lceil 2\log n\rceil$ であり，返答内容の大きさは $(2^{c-1} + 4\log m)\cdot\lceil 2\log n\rceil$ である．問い合わせ内容に含まれる $\log m\cdot\lceil 2\log n\rceil$ および返答内容に含まれる $4\log m\cdot\lceil 2\log n\rceil$ は，$(1,m)$-HCOT と同様，参加者がプロトコルに従わない malicious player であっても安全性要件を満た

すために必要な暗号文の大きさである．この暗号文を生成する処理は，$(1,m)$-HCOT と同様であるため説明を省略する．

5.2 問い合わせ内容の変換

提案するデータ検索プロトコルでは，検索結果を得るための問い合わせ内容 $Q(i)$ を Merkle 木の各深さ $h \in [\log N]$ における証明値の要素 $p(i,D)_h$ を得るための問い合わせ内容（以降では $Q(i,h)$ と表す）に変換する．この変換は，**4.2** で述べたように，深さ $\log N$ の完全 2 分木の葉節点 $x_{\log N,i}$ に対する問い合わせ内容を，その葉節点の各先祖節点 $x' \in TA(x_{\log N,i})$ に対する問い合わせ内容に変換できれば実現できる．また，葉節点に対する問い合わせ内容から，その先祖節点に対する問い合わせ内容への変換は，節点 $x_{h,i}$ に対する問い合わせ内容をその親節点 $x_{h-1,\lceil i/2 \rceil}$ に対する問い合わせ内容に変換できれば実現できる．すなわち，$Q(i,h)$ である，$(1,2^h)$-TOT における検索語 i に対する問い合わせ内容から，$Q(i,h-1)$ である，$(1,2^{h-1})$-TOT における検索語 $\lceil i/2 \rceil$ に対する問い合わせ内容を生成できれば良い．

本稿では，$(1,2^h)$-TOT-$Q(i)_{\langle l_1,\ldots,l_c,c\rangle}$ に対して，$l_c = 2$ であれば $(1,2^{h-1})$-TOT-$Q(\lceil i/2 \rceil)_{\langle l_1,\ldots,l_{c-1},c-1\rangle}$ を，$l_c > 2$ であれば $(1,2^{h-1})$-TOT-$Q(\lceil i/2 \rceil)_{\langle l_1,\ldots,l_c/2,c\rangle}$ を生成する変換法を提案する．$Q(i,h)$ は，c 組の暗号文 $\langle E(I(t,i_j)))_{t\in[l_j],j\in[c]}$ である．

（1） $l_c = 2$ の場合の変換

$Q(i,h-1)$ は，$Q(i,h)$ から c 組目の暗号文 $\langle E(I(t,i_c)))_{t\in[l_c]}$ を除くことで生成できる．これは，$(l_1,\ldots,l_c)$-木における $x_{h,i}$ の位置情報 $\langle i_1,\ldots,i_c\rangle$ から i_c を除いた $\langle i_1,\ldots,i_{c-1}\rangle$ は，$(l_1,\ldots,l_{c-1})$-木における $x_{h-1,\lceil i/2 \rceil}$ の位置情報となるためである．

（2） $l_c > 2$ の場合の変換

$Q(i,h-1)$ は，$Q(i,h)$ の c 組目の暗号文 $\langle E(I(t,i_c)))_{t\in[l_c]}$ に対して，以下の計算を行って得られる $l_c/2$ 個の暗号文を代わりに用いることで生成できる．

$$E(I(2t-1,i_c)) \cdot E(I(2t,i_c)) \quad (t \in [l_c/2])$$

ここで，元の l_c 個の暗号文は i_c 番目が 1 を，他は 0 を暗号化したものであるため，この式で生成される $l_c/2$ 個の暗号文は $\lceil i_c/2 \rceil$ 番目が 1 を，他は 0 を暗号化したものとなる．すなわち，次の式が成り立つ．

$$E(I(2t-1,i_c)) \cdot E(I(2t,i_c)) = E(I(t,\lceil i_c/2 \rceil)) \quad (t \in [l_c/2])$$

これを代わりに用いるのは，$(l_1,\ldots,l_c)$-木における $x_{h,i}$ の位置情報 $\langle i_1,\ldots,i_c\rangle$ の i_c を $\lceil i_c/2 \rceil$ に変更した $\langle i_1,\ldots,\lceil i_c/2 \rceil\rangle$ は，$(l_1,\ldots,l_c)$-木における $x_{h-1,\lceil i/2 \rceil}$ の位置情報となるためである．

5.3 提案するデータ検索プロトコルにおける処理

提案するデータ検索プロトコルにおける各処理を示す．返答内容の生成処理は，検索結果に関する返答内容の生成，問い合わせ内容の変換，証明値に関する返答内容の生成の三つの部分処理からなる．提案するデータ検索プロトコルでは，Merkle 木と $(1,m)$-TOT を用いる．なお，ユーザが問い合わせ内容の生成で用いる $(1,m)$-TOT のパラメータ $l_1,\ldots,l_c,c$ は予めユーザとサーバの間で合意しておくものとする．

[コミット値と証明値の生成] **4.1** で示した Merkle 木の利用法に従い，コミット値と証明値を生成する．

[問い合わせ内容の生成] 検索語 i に対応した検索結果 d_i を得るための問い合わせ内容 $Q(i)$ を生成する．$(1,N)$-TOT において問い合わせ内容の生成処理を実行し，生成した TOT-$Q(i)_{\langle l_1,\ldots,l_c,c\rangle}$ を問い合わせ内容 $Q(i)$ とする．

[返答内容の生成] 以下の三つの処理を実行し，検索結果に関する返答内容と証明値に関する返答内容を生成し，その二つを合わせて返答内容 $A(Q(i),D)$ とする．

- **[検索結果に関する返答の生成]**

問い合わせ内容 $Q(i)$ とデータベース D に対して検索結果に関する返答を生成する．具体的には，$(1,N)$-TOT の返答内容の生成処理において，入力をデータベース $D = \{d_1,\ldots,d_N\}$，問い合わせ内容 $Q(i)$，パラメータ $(l_1,\ldots,l_c,c)$ として実行する．生成された TOT-$A(Q(i),D)_{\langle l_1,\ldots,l_c,c\rangle}$ を $A(Q(i))$ と表し，検索結果に関する返答内容とする．

- **[問い合わせ内容の変換]**

5.2 で示した問い合わせ内容の変換を実行し，検索結果 d_i に関する問い合わせ内容 $Q(i)$ を証明値の各要素 $p(i,D)_h$（$h \in [\log N]$）に関する問い合わせ内容 $Q(i,h)$ に変換する．ここで，得られた $Q(i,h)$ を用いる $(1,2^h)$-TOT のパラメータを $(l_{h,1},\ldots,l_{h,c_h},c_h)$ と表す．

- **[証明値に関する返答の生成]**

問い合わせ内容の変換処理で生成した証明値に関する問い合わせ内容 $\{Q(i,h) \mid h \in [\log N]\}$ と Merkle 木の深さ h における 2^h 個の節点の値に対して証明値に関する返答を生成する．具体的には，各 $h \in [\log N]$ に対して入力を $D_h = \{d_{h,j} = V(\overline{x_{h,j}}) \mid j \in [2^h]\}$，問い合わせ内容 $Q(i,h)$，パラメータ $(l_{h,1},\ldots,l_{h,c_h},c_h)$ として実行する．生成された TOT-$A(Q(i,h),D)_{\langle l_{h,1},\ldots,l_{h,c_h},c_h\rangle}$ を $A(Q(i,h))$ と表し，$\{A(Q(i,h)) \mid h \in [\log N]\}$ を証明値に関する返答内容とする．

[検索結果の入手] 返答内容の内，検索結果に関する返答内容 $A(Q(i))$ を用いて $(1,N)$-TOT における検索結果の入手処理を実行し，検索結果 d_i を得る．

[証明値の入手] 返答内容の内，証明値に関する返答内容 $\{A(Q(i,h)) \mid h \in [\log N]\}$ を用いて $(1,2^h)$-TOT における検索結果の入手処理を実行し，証明値の各要素 $p(i,D)_h$ を得る．そして，$\{p(i,D)_h \mid h \in [\log N]\}$ を証明値 $p(i,D)$ とする．

[検索結果の一貫性検証] 検索結果 d_i と証明値 $p(i,D)$ を用いて，**4.1** で示した方法に従い検索結果の一貫性を検証する．

6. 評　価

提案プロトコルの安全性と効率をそれぞれ評価する．

6.1 安 全 性

サーバおよびユーザが malicious player であっても提案プロトコルが **2.2** の安全性に関する要件を満たすことを，TOT の安全性の仮定の下で示す．なお，HCOT [4] はサーバが malicious player であっても要件 1 を満たし，ユーザが malicious player であっても要件 2 を満たすことが [4] で示されている．TOT は

HCOT の自然な拡張であり，HCOT と同様に安全性を証明できるため，ここではその証明は省略する．

要件 1（検索語の秘匿） サーバがユーザから入手できる情報は，TOT における問い合わせ内容だけである．サーバば malicious player であっても検索語の情報を得られないことは TOT の安全性より保証される．よって要件 1 を満たす．

要件 2（データベースの秘匿） ユーザがサーバから入手できる情報は，検索結果に関する返答内容，証明値に関する返答内容，コミット値である．検索結果に関する返答内容は TOT の実行によって入手する情報であり，ユーザが malicious player であっても検索結果から分かること以外のデータベースの情報が得られないことは TOT の安全性より保証される．一方，証明値に関する返答内容とコミット値は，データベース内の各データに対して衝突困難一方向性ハッシュ関数を適用したものであり，ハッシュ関数の安全性より，ユーザが検索結果から分かること以外のデータベースの情報を得ることはできない．よって，要件 2 を満たす．

要件 3（検索結果の一貫性検証） ハッシュ関数の衝突困難性より，サーバは Merkle 木を利用して生成したコミット値（根節点の値）を変更することなく葉節点の値を変更できない．また，葉節点の値はデータベースの各データに衝突困難一方向性ハッシュ関数を適用した値であり，同様に葉節点の値を変更することなくデータベースの各データを変更できない．よって，提案プロトコルは要件 3 を満たす．

6.2 効　　率

通信効率として，問い合わせ内容，返答内容，コミット値および証明値の大きさを評価する．また，計算量として，サーバおよびユーザが実行するべき乗剰余演算の回数を評価する．比較対象として，[12] のプロトコルと [7] の OKS を挙げる．なお，OKS は，本稿で対象としているデータ検索モデルを含むモデルを対象としている．よって，OKS を本稿のモデルに合わせた場合の通信量と計算量を比較する．また，提案プロトコルおよび [12] のプロトコルは，通信効率に関するパラメータをもつが，そのパラメータを通信効率に関する要件を満たしながら返答内容の大きさを最小とする値（$c = 2$）とする．

表 2　通信量の比較

	提案プロトコル	[12] のプロトコル	OKS [7]
問い合わせ内容	$O(\sqrt{N})$	$O(\sqrt{N})$	$O(1)$
返答内容	$O((\log N)^2)$	$O(N)$	$O(1)$
コミット値	$O(1)$	$O(1)$	$O(N)$
証明値	$O(\log N)$	$O(\log N)$	$O(1)$

表 3　計算量の比較

	提案プロトコル	[12] のプロトコル	OKS [7]
ユーザ	$O(\sqrt{N})$	$O(\sqrt{N})$	$O(1)$
サーバ	$O(\sqrt{N}\log N)$	$O(N\log N)$	$O(1)$

提案プロトコルでは，**5.1** で新たに提案した TOT を用いることで，[12] のプロトコルに比べて問い合わせ内容の大きさやユーザの計算量を増やすことなく，返答の通信量とサーバの計算量を削減した．提案プロトコルはコミット値の大きさがデーターベースの大きさに依存しないため，OKS に比べてデータベースの内容が頻繁に更新されるときに有効となる．

7. まとめと今後の課題

本稿では，データ検索プロトコルとして，安全性と通信効率に関する要件を全て満たしたプロトコルを提案した．安全性に関する要件は，検索語の秘匿，データベースの秘匿，検索結果の一貫性検証である．通信効率に関する要件は，問い合わせ内容，返答内容，コミット値，証明値の大きさがそれぞれデータベースの大きさよりも小さいことである．Merkle 木と効率の良い Oblivious Transfer (OT) を組み合わせることでこれらの要件を満たすことができる．本稿では，[12] と同様，通信効率を向上させるため，検索結果を得るための問い合わせ内容を証明値を得るための問い合わせ内容にサーバが変換して再利用する．これにより，問い合わせの通信量を検索結果を得るための問い合わせ内容だけに削減した．更に，変換によって返答内容の大きさを増加させないようにするため，[4] で提案されている OT を拡張して新たに効率の良い OT を構成し，これを用いることで [12] のプロトコルに比べて返答の通信量やサーバの計算量を削減した．

今後の課題として，通信量や計算量の更なる削減や，別のデータ検索モデルへの提案プロトコルの適用が挙げられる．

文　　献

[1] C.J. van Rijsbergen, Information Retrieval, Butter-worths, 1979.

[2] A. Buldas, P. Laud, and H. Lipmaa, "Accountable Certificate Management using Undeniable Attestations," ACM CCS-7, pp.9–18, 2000.

[3] A. Buldas, M. Roos, and J. Willemson, "Undeniable Replies for Database Queries," Fifth International Baltic Conference on Database and Information Systems, vol.2, pp.215–226, 2002.

[4] Y.C. Chang, "Single Database Private Information Retrieval with Logarithmic Communication," ACISP2004, LNCS3108, pp.50–61, 2004.

[5] Y. Gertner, Y. Ishai, E. Kushilevitz, and T. Malkin, "Protecting Data Privacy in Private Information Retrieval Schemes," STOC'98, pp.151–160, 1998.

[6] S. Micali, M. Rabin, and J. Killian, "Zero-Knowledge Sets," FOCS'03, pp.80–91, 2003.

[7] W. Ogata and K. Kurosawa, "Oblivious Keyword Search," J. of Complexity, vol.20, pp.356–371, 2004.

[8] P. Paillier, "Public-key cryptosystems based on composite degree residuosity classes," EUROCRYPTO'99, LNCS1592, pp.223–238, 1999.

[9] P. Fouque, G. Poupard, and J. Stern, "Sharing Decryption in the Context of Voting or Lotteries," Financial Cryptography 2000, LNCS1962, pp.90–104, 2001.

[10] R.C. Merkle, "Protocols for Public Key Cryptosystems," IEEE Symposium on Security and Privacy, pp.122–134, 1980.

[11] M. Szydlo, "Merkle Tree Traversal in Log Space and Time," EUROCRYPT'04, LNCS3027, pp.541–554, 2004.

[12] 中山 敏, 藤原 晶, 吉田 真紀, 藤原 融, "検索結果の秘匿と一貫性検証を可能とするデータ検索プロトコルの提案," SCIS2005 予稿集, Vol.III, pp.1489–1494, 2005.