加法準同型暗号を用いた化合物データベースの秘匿検索プロトコル

縫田 光司† 清水 佳奈‡ 荒井 ひろみ§ 浜田 道昭$ 津田 宏治‡
広川 貴次‡ 花岡 悟一郎† 佐久間 淳¡ 浅井 潔‡

†産業技術総合研究所セキュアシステム研究部門
305-8568 茨城県つくば市梅園 1-1-1 中央第 2
k.nuida[at]aist.go.jp（縫田）

‡産業技術総合研究所生命情報工学研究センター §理化学研究所生命情報基盤研究部門
$ 東京大学大学院新領域創成科学研究科 ¡筑波大学大学院システム情報工学研究科

あらまし あるユーザが検索目的で化合物データベースの購入を検討する際、ユーザ側は検索対象の化合物の情報をサーバ（データベース販売者）側に漏らしたくないが、サーバ側も検索対象の化合物の有無以外の余分な情報をユーザ側に漏らしたくない、という状況が考えられる。本発表では、ユーザ側だけでなくサーバ側の入力情報の秘匿も考慮し、さらに加法準同型暗号を用いた工夫により計算コストと通信ラウンド数にも配慮した、化合物の類似度判定プロトコルを提案する。

Privacy-preserving database search protocol for chemical compounds with additive-homomorphic encryption

NUIDA, Koji† SHIMIZU, Kana‡ ARAI, Hiromi§ HAMADA, Michiaki$
TSUDA, Koji‡ HIROKAWA, Takatsugu‡ HANAOKA, Goichiro†
SAKUMA, Jun¡ ASAI, Kiyoshi‡

†Research Institute for Secure Systems (RISEC),
National Institute of Advanced Industrial Science and Technology (AIST)
AIST Tsukuba Central 2, 1-1-1 Umezono, Tsukuba, Ibaraki 305-8568, Japan
k.nuida[at]aist.go.jp (NUIDA)

‡Computational Biology Research Center (CBRC), AIST

§Bioinformatics And Systems Engineering (BASE) division, RIKEN

$ Graduate School of Frontier Sciences, The University of Tokyo

¡Graduate School of SIE, University of Tsukuba

Abstract When a user considers to buy a chemical compound database for searching, it is a natural situation that the user would not like to leak any information on the target object to the server (database seller), while the server also would not like to leak any information on the database other than the (in)existence of the target object. In this talk, we propose a protocol for calculating similarities between chemical compounds, which concerns concealment of information of not only the user but also the server, and which aims at reducing computational and communication costs by making use of additive-homomorphic encryption schemes.

1　研究の背景と成果のあらまし

近年、創薬などの分野で所望の性質を持つ化合物を探求する際に、既存の化合物データベースの中から目標の化合物に近い性質を持つ化合物を探してそのデータを利用することが行われている。クライアントが市販の化合物データベースの購入を検討する際、自分が求める種類の化合物がどのくらい含まれているかを事前に確認したいというのは自然な要求であろう。このとき、クライアントの検索目標となる化合物の情報には、研究開発の鍵となるアイデアや個人の疾患に関連する情報をはじめ秘匿性の高い情報が含まれ得るため、データベースの確認の際に検索の詳細な内容をデータベース側に教えることは一般に歓迎されない。このクライアント側の要求を満たすだけであれば、データベースの全部もしくは一部をクライアント側に渡して中身を確認させる方法も考えられる。しかし、一方のデータベース販売側としては、売買が成立する前の段階では、多大なコストを払って構築したデータベースの情報をなるべくクライアントに漏らしたくないと考えるであろう。この両者の相反する要求をできる限り両立させつつ、クライアントの検索目標の化合物とデータベース内の化合物との類似度を判定する手法の確立が望まれる。クライアント側の検索内容の情報をデータベース側に対して秘匿しつつ、データベースに関する情報も「検索内容と類似する収録データの個数」以外は何もクライアント側に漏らさない状態が双方にとって理想的である。

化合物データベースで用いられる標準的な化合物データ符号化法の下で、二つの化合物の類似度の定量的指標の一つに Jaccard index がある [5]。クライアント側の検索目標である化合物の情報を漏らさずにデータベース側が持つ化合物との Jaccard index を計算する既存方式としては、Singh ら [8]、Blundo ら [1]、Zhang ら [10] の方式などが知られている。これらの方式において最終的にクライアント側が得る情報はデータベース側の化合物との Jaccard index の具体的な値である。しかし、検索目標と類似する化合物の有無や個数を知るという目的からすると、Jaccard index の値がある閾値より大きいかどうかだけをクライアント側に知らせれば充分であり、Jaccard index の正確な値を知らせることはデータベース側の情報を必要以上に漏らしてしまうことに繋がるという懸念がある。そのため、Jaccard index の具体的な値ではなくその閾値との大小関係のみをクライアント側に送信する方式の構築が望まれる。

本研究では以上の問題に取り組み、クライアント側だけでなくデータベース側の情報秘匿にも配慮した、化合物データベースの類似度検索プロトコルを考案した。このプロトコルにおいては、クライアント側の検索内容を暗号化した状態で取り扱うため、強秘匿性（ないし選択平文攻撃に対する識別不可能性）を持つ公開鍵暗号方式を用いることで検索内容の秘匿を達成できる。一方のデータベース側の化合物情報の秘匿については、Jaccard index の値そのものではなく、Jaccard index と関連付けられるある値（本稿では「閾値型 Jaccard index」と呼ぶ）をクライアント側に送信し、その値の正負によって化合物同士の「類似性が高い」かどうかを表す方法を採っている。Jaccard index 自体の計算は加減算の他に除算を必要とするが、この閾値型 Jaccard index は加減算だけで計算できるため、加法準同型暗号（例えば [6] や、[3] において乗法巡回群の元の指数を平文と看做したもの）を用いれば、通信ラウンド数の大きい複雑なプロトコルや信頼できる第三者を利用せずとも暗号状態での計算が可能である。この点が本研究成果の中心的アイデアの一つである。なお、本提案方式は実際には、Jaccard index に留まらずその一般化である Tversky index[9] の利用にも適応しており、本稿では Tversky index を用いた形でプロトコルを記述する。

実は、閾値型 Jaccard index もデータベース側の化合物情報を（所望の大小関係以外）完全に秘匿できてはおらず、Jaccard index の値自体を返す場合よりは小さいながらもある程度の情報をクライアント側に漏らしてしまう。この漏れる情報量の削減のため、閾値型 Jaccard index と一緒に複数のダミー値をクライアント側に送信する方法や、閾値型 Jaccard index をさらにランダムに変形する方法についても本稿で論じる。

表 1: 既存方式と提案方式の比較

方式	出力	通信量	ラウンド数	計算量	TPP
本方式	**JI の大小（閾値型 JI）**	**小**	**1**	**小**	**なし**
Singh et al. [8]	JI	小	> 1	小	必要
EsPRESSo [1]	JI	小	1	小	なし
Zhang et al. [10]	JI	小	1	小	なし
多者計算（一般）	JI の大小	大	> 1	大	なし
完全準同型暗号 [4]	JI の大小	大	1	大	なし

（JI：Jaccard index、TPP：信頼できる第三者）

表 1 では本提案方式と既存方式との性能比較を行っている。既存方式では Jaccard index の値自体をクライアント側に送信する仕様になっており、Jaccard index の値自体の送信を避けることでデータベース側の情報秘匿にも配慮した方式は著者らの知る限り本提案方式が初めてである。(理論的には、多者計算プロトコルの一般論や完全準同型暗号 [4] を用いることで Jaccard index の大小の情報のみを返す類似度検索プロトコルの構成が可能である。しかし、少なくとも現時点では、これらの方式を実際のアプリケーションに適用するのは効率の面でおよそ現実的ではないと考えられる。)

なお、一般的な化合物データベースにおいては化合物をビット列として符号化する方法を採用しているため、本提案方式でクライアント側からデータベース側に暗号化して送信される検索内容（平文）もこの符号化に則ったビット列であると仮定している。しかし、クライアント側が悪意ある攻撃者である場合、あえて本来の符号化に則らない異常値を暗号化して送信することにより、データベース側の情報を大きく抜き取ろうとすることも考えられる。この攻撃への対応策として、坂井ら [7] によって提案された、ある暗号文に対応する平文が所定の範囲から選ばれたものであることを証明できる非対話ゼロ知識証明の利用が考えられる。クライアント側からの暗号文にこの非対話ゼロ知識証明を添えておくことで、データベース側は暗号文に秘められた異常値を前もって検知することができる。なお、ゼロ知識証明なので、この対応策を講じたとしても（善良な）クライアント側の情報秘匿に影響は出ないことを注意しておく。

本稿の構成は以下の通りである。2 章では、化合物の類似度評価の指標である Jaccard index および Tversky index（2.1 節）、準同型暗号（2.2 節)、非対話ゼロ知識証明（2.3 節）といった諸概念の導入と整理を行う。3 章では本研究での提案プロトコルを述べた（3.1 節）上で、クライアント側の安全性について論じ（3.2 節)、さらにデータベース側の安全性を増強するための方策を提案する（3.3 節)。4 章では、本提案方式の安全性について定量的評価を行う。より理論的な安全性評価は今後の課題とする。最後に 5 章において、悪意あるクライアントの異常値を用いた攻撃への対策について述べる。

2 準備

2.1 化合物の類似度指標

一般的な化合物データベースにおいて化合物の情報を保存する際、ある種の部分構造の有無など化合物を特徴付ける性質を予めいくつか挙げておき、それらの性質の各々を有する（1）か有しない（0）かの 1 ビット情報の列（これを化合物の「**フィンガープリント**」と呼ぶ）として化合物を符号化している。上で挙げられた性質の総数が ℓ であれば、フィンガープリントは ℓ ビット列 $\vec{p} = (p_i)_{i=1}^{\ell} \in \{0,1\}^{\ell}$ で表される（以下、ビット列 $\vec{p}$ を、$p_i = 1$ となる添字 i からなる $\{1, 2, \ldots, \ell\}$ の部分集合としばしば同一視する)。二つのフィンガープリント $\vec{p}, \vec{q}$ の関数

であって、元々の化合物同士の類似度をよく近似すると認められている指標の一つが Tversky index である [9]。Tversky index $\mathsf{TI}(\vec{p},\vec{q})$ は、実数 $\alpha,\beta\in[0,1]$ をパラメータとして持ち、以下のように定義される：

$$\mathsf{TI}(\vec{p},\vec{q}) := \frac{|\vec{p}\cap\vec{q}|}{|\vec{p}\cap\vec{q}|+\alpha|\vec{p}\setminus\vec{q}|+\beta|\vec{q}\setminus\vec{p}|}\ .$$

TI の値がある閾値以上のとき、$\vec{p}$ と $\vec{q}$ の元となる化合物同士は「似ている」と判断する。パラメータを $\alpha=\beta=1$ と設定した場合の指標は Jaccard index と呼ばれる [5]。本稿で考察する問題の基本形は、クライアント（検索者）側とデータベース側がそれぞれフィンガープリント $\vec{p}$ と $\vec{q}$ を指定したとき、互いの p と q の情報を極力漏らさないようにしつつ、$\vec{p}$ と $\vec{q}$ の元となる化合物同士が上記の意味で「似ている」か否かをクライアントに教えるという問題である。

2.2 準同型暗号

定義 1. 本稿では、以下の機能を持つアルゴリズムの組 $(\mathsf{Gen},\mathsf{Enc},\mathsf{Dec},\mathsf{Add},\mathsf{Inv})$ を（**加法**）**準同型暗号方式**と呼ぶ。

鍵生成 鍵生成アルゴリズム Gen はセキュリティパラメータ 1^k を入力として、公開鍵 pk と秘密鍵 sk の対を出力する。対応する平文空間（加法群）を M、暗号文空間を C で表す。

暗号化 暗号化アルゴリズム Enc は公開鍵 pk と平文 $m\in M$ を入力として暗号文 $c\in C$ を出力する：$c\leftarrow\mathsf{Enc}(\mathsf{pk},m)$。

復号 復号アルゴリズム Dec は秘密鍵 sk と暗号文 $c'\in C$ を入力として平文 $m'\in M$ もしくは $\perp$ を出力する[1]：m' or $\perp\leftarrow\mathsf{Dec}(\mathsf{sk},c')$。さらに、$\mathsf{Dec}(\mathsf{sk},\mathsf{Enc}(\mathsf{pk},m))=m$ が確率 1 で成り立つ（健全性）。

準同型演算 加算アルゴリズム Add は公開鍵 pk と暗号文 $c_1,c_2\in C$ を入力として暗号文 $c\in C$ もしくは $\perp$ を出力する[2]。一方、符号反転アルゴリズム Inv は公開鍵 pk と暗号文 $c\in C$ を入力として暗号文 $c'\in C$ もしくは $\perp$ を出力する。さらに、

$$\mathsf{Dec}(\mathsf{sk},\mathsf{Add}(\mathsf{pk},\mathsf{Enc}(\mathsf{pk},m_1),\mathsf{Enc}(\mathsf{pk},m_2))) = m_1+m_2\ ,$$
$$\mathsf{Dec}(\mathsf{sk},\mathsf{Inv}(\mathsf{pk},\mathsf{Enc}(\mathsf{pk},m))) = -m$$

がそれぞれ確率 1 で成立する。

アルゴリズム Add を繰り返し用いることで、暗号文 c の n 個の和（n は非負整数）を計算することができる。この結果を簡略化のために $n\cdot c$ と書く[3]。また、$n<0$ のときには、$n\cdot c$ は $\mathsf{Inv}(\mathsf{pk},|n|\cdot c)$ のことであると定める[4]。

特に本稿では、平文空間が加法巡回群である方式を用いる。例えば、Paillier 暗号 [6] や、ElGamal 暗号 [3] において乗法群の元の指数部を平文とみたものはこの条件を満たす。

2.3 非対話ゼロ知識証明

ゼロ知識証明とは、ある命題が成り立つことを知っている証明者が、「その命題が成り立つ」事実以外の一切の情報を漏らすことなく、検証者に「その命題が成り立つ」ことを納得させるためのプロトコルである。特に、このプロトコルで検証者から証明者への通信が行われない（つまり、証明者が一方的に証拠を提示して、検証者はそれを手元で確認する）なら、**非対話ゼロ知識証明**と呼ばれる。例えば、ElGamal 暗号などある種の公開鍵暗号方式については、「この暗号文に対応する平文は所定の範囲から選ばれたものである」という命題の非対話ゼロ知識証明が構成されている [7]。

3 提案方式

本章では、2.1 節で言及したように、Alice（クライアント）と Bob（データベース）の指定し

[1] $\perp$ は暗号文が不正である場合に相当する。

[2] $\perp$ は不正な暗号文を入力した場合に相当する。アルゴリズム Inv についても同様。

[3] $0\cdot c$ は 0 を暗号化した結果であるとする。

[4] 大抵の方式では、$(\mathrm{ord}(m)-1)\cdot c$（ここで $\mathrm{ord}(m)$ は c に対応する平文 m の（加法的）位数を表す）の計算によって $\mathsf{Inv}(c)$ の計算に代えることができるが、Add の繰り返しよりも効率的に Inv を計算できる場合もあるので、冗長に見えても Inv を用いた定式化を採用している。

たフィンガープリント $\vec{p}$ と $\vec{q}$ について、互いの $\vec{p}$ と $\vec{q}$ の情報を極力漏らさないようにしつつ、Tversky index $\mathsf{TI}(\vec{p},\vec{q})$ が与えられた閾値 $\theta \in [0,1]$ 以上であるかを Alice に教える問題を扱う。

3.1 秘匿検索プロトコルの概要

Tvarsky index のパラメータ α, β と閾値 θ を有理数と仮定し、$\alpha = \mu_\mathrm{a}/\gamma$、$\beta = \mu_\mathrm{b}/\gamma$、$\theta = \theta_\mathrm{n}/\theta_\mathrm{d}$ とそれぞれ既約分数表示しておく。すると、$\mathsf{TI}(\vec{p},\vec{q}) \geq \theta$ という条件は

$$\frac{|\vec{p}\cap\vec{q}|}{|\vec{p}\cap\vec{q}| + (\mu_\mathrm{a}/\gamma)|\vec{p}\setminus\vec{q}| + (\mu_\mathrm{b}/\gamma)|\vec{q}\setminus\vec{p}|} \geq \frac{\theta_\mathrm{n}}{\theta_\mathrm{d}}$$

と書き表される。両辺の分母を払った上で関係式 $|\vec{p}\setminus\vec{q}| = |\vec{p}| - |\vec{p}\cap\vec{q}|$、$|\vec{q}\setminus\vec{p}| = |\vec{q}| - |\vec{p}\cap\vec{q}|$ を用いて整理すると、件の条件は

$$\overline{\mathsf{TI}}(\vec{p},\vec{q}) := \Gamma|\vec{p}\cap\vec{q}| - \theta_\mathrm{n}(\mu_\mathrm{a}|\vec{p}| - \mu_\mathrm{b}|\vec{q}|) \geq 0$$
$$\text{ただし } \Gamma := (\theta_\mathrm{d} - \theta_\mathrm{n})\gamma + \theta_\mathrm{n}(\mu_\mathrm{a} + \mu_\mathrm{b})$$

と同値であることがわかる。$\overline{\mathsf{TI}}(\vec{p},\vec{q})$ を $\vec{p}$ と $\vec{q}$ の**閾値型 Tversky index** と呼ぶ。

以下では平文空間が加法巡回群 $\mathbb{Z}/N\mathbb{Z}$ である加法準同型暗号方式（具体例は 2.2 節を参照）を用いて、$\overline{\mathsf{TI}}(\vec{p},\vec{q})$ の値を暗号化して取り扱う。その際、$\mathbb{Z}/N\mathbb{Z} = \{0,1,\ldots,N-1\}$ と同一視した上で、先頭から半分の範囲 $\{0,1,\ldots,\lfloor N/2\rfloor\}$ にある平文を「正の数」、それ以外の平文を「負の数」と看做すことにする。このとき、本来は正の値である $\overline{\mathsf{TI}}(\vec{p},\vec{q})$ が桁溢れによって「負の数」と誤判定されることがないよう、N を充分大きな値にしておく必要がある。

以上を踏まえて、Alice と Bob の二者間プロトコルを提案する。Alice と Bob は、それぞれ ℓ ビット列 $\vec{p}$ と $\vec{q}$ を秘密に保持しているとする。

1. Alice は鍵生成アルゴリズム $\mathsf{Gen}(1^k)$ を実行し鍵対 $(\mathsf{pk},\mathsf{sk})$ を得て、pk を公開する。
2. Alice は $\vec{p}$ の各成分 $p_i \in \{0,1\}$ を鍵 pk で暗号化し、暗号文 $c_{\mathrm{A},i} \leftarrow \mathsf{Enc}(\mathsf{pk}, p_i)$（$1 \leq i \leq \ell$）を Bob に送信する。
3. Bob は鍵 pk とアルゴリズム Add を用いて、$q_i = 1$ である添字 i の全てに対する暗号文 $c_{\mathrm{A},i}$ の和 $c_{\mathrm{B},\cap}$ を計算する[5]。同様に、全ての添字 $1 \leq i \leq \ell$ に対する暗号文 $c_{\mathrm{A},i}$ の和 $c_{\mathrm{B,p}}$ を計算する。さらに、$|\vec{q}|$ の暗号文を $c_{\mathrm{B,q}}$ とする。
4. Bob は暗号文 c_TI を
$$c_\mathrm{TI} := \Gamma \cdot c_{\mathrm{B},\cap} - \theta_\mathrm{n} \cdot (\mu_\mathrm{a} \cdot c_{\mathrm{B,p}} - \mu_\mathrm{b} \cdot c_{\mathrm{B,q}})$$
で計算して Alice に送信する（和の計算には Add を用いている）。
5. Alice は鍵 sk を用いて c_TI を復号し、$T \in M$ を得る：$T \leftarrow \mathsf{Dec}(\mathsf{sk}, c_\mathrm{TI})$。$T$ が上記の意味で「正の数」であれば Alice は 1 を出力し、そうでなければ 0 を出力する。

準同型暗号方式の性質より、$c_{\mathrm{B},\cap}, c_{\mathrm{B,p}}, c_{\mathrm{B,q}}$ はそれぞれ $|\vec{p}\cap\vec{q}|, |\vec{p}|, |\vec{q}|$ の暗号文であり、$T = \overline{\mathsf{TI}}(\vec{p},\vec{q})$ が成り立つ。よって、このプロトコルにより、Alice は確かに $\mathsf{TI}(\vec{p},\vec{q}) \geq 0$ かそうでないかを知ることができる。

3.2 クライアント側の安全性

Alice から Bob への通信内容は、公開鍵および暗号文 $c_{\mathrm{A},1},\ldots,c_{\mathrm{A},\ell}$ のみである。準同型暗号方式が強秘匿性（ないし選択平文攻撃に対する識別不可能性）を持つならば、これらの通信内容から Alice の持つフィンガープリント $\vec{p}$ の情報が（計算量的に）何も漏れないようにできる。

3.3 データベース側の安全性増強手法

Alice が攻撃者であるとき、理想的には、上のプロトコルで $\vec{p} \in \{0,1\}^\ell$ をどのように設定しても、Bob の持つフィンガープリント $\vec{q}$ についての情報が、$\mathsf{TI}(\vec{p},\vec{q}) \geq 0$ か否かだけしか Alice に漏れないことが望ましい。しかし現在のプロトコルでは、Tversky index そのものではないにせよ、閾値型 Tversky index の値 $T = \overline{\mathsf{TI}}(\vec{p},\vec{q})$ を Alice に渡してしまっているため、余分な情報が漏れてしまっていると考えられる。その情報の漏れを極力少なくするために、以下の方針で提案プロトコルに改良を加える。

[5] $q_i = 1$ である添字 i が一つもないときには、0 を暗号化した結果を $c_{\mathrm{B},\cap}$ とする。

ダミー情報の追加　Bob が Alice に暗号文 c_{TI} を送信する際、ある確率分布に従って選んだダミー値 $T'_1, \ldots, T'_d$ の暗号文 $c'_1, \ldots, c'_d$ を一緒に（順番をランダムに並び替えて）送信し、併せて「T'_i たちの中で「正の値」が何個あるか」（この個数を $N_{\mathrm{p,dummy}}$ と書く）を Alice に送信する。Alice は受け取った暗号文を全て復号し、「正の値」の個数 $N_{\mathrm{p,all}}$ を数える。このとき、$N_{\mathrm{p,all}} = N_{\mathrm{p,dummy}} + 1$ であれば $\mathsf{TI}(\vec{p}, \vec{q}) \geq 0$、そうでなければ $\mathsf{TI}(\vec{p}, \vec{q}) < 0$ と正しく判定することができるため、プロトコルの正しさは損なわれない。一方安全性について定性的な観点からは、Alice が得た（一般には複数の）「正の値」の中でどれが本物の閾値型 Tversky index であるかがダミー値の存在によって隠され、直ちにはわからない状態となっていることから、Alice に漏れる $\vec{q}$ の情報量が削減できていると考えられる。さらに、ダミー値の個数 d を大きくすることで、Alice に漏れる余分な情報量を限りなく 0 に近付けることができる。より定量的な安全性評価については 4 章で述べる。

類似度のアフィン変換　Bob が Alice に暗号文 c_{TI} を送信する段階で、ある確率分布に従って選んだ整数 $r > 0$ と $0 \leq s \leq r-1$ を用いて新たな暗号文 $\tilde{c}_{\mathsf{TI}} := \mathsf{Add}(\mathsf{pk}, r \cdot c_{\mathsf{TI}}, \mathsf{Enc}(\mathsf{pk}, s))$（これは定義から $r\overline{\mathsf{TI}}(\vec{p}, \vec{q}) + s$ の暗号文となる）を作り、それを c_{TI} の代わりに Alice へ送信する。このとき、r と s の範囲の選び方より、$\overline{\mathsf{TI}}(\vec{p}, \vec{q}) \geq 0$ と $r\overline{\mathsf{TI}}(\vec{p}, \vec{q}) + s \geq 0$ とは同値なので、この変更によってプロトコルの正しさが損なわれることはない[6]。また安全性については、Alice に閾値型 Tversky index の値そのものではなく値をある程度ランダムに変換した結果を渡しているので、元々のプロトコルよりも Alice に漏れる情報量がより小さくなる。

さらに、前の段落で述べたダミー情報の利用と組み合わせる（このアフィン変換を考慮に入れてダミー値の分布を決める）ことで、漏れる情報量の削減効果を強められると期待される。

[6]ただし、$\overline{\mathsf{TI}}(\vec{p}, \vec{q})$ や r や s が最大の場合でも桁溢れが起きないよう、N を充分大きく選んでおく必要がある。

複数フィンガープリントのランダム置換　上記のプロトコルでは Bob の持つフィンガープリントは一つだけであるが、一般にはデータベースには複数の化合物データが収められているため、対応するフィンガープリントも複数となる。この状況においては、Alice の指定するフィンガープリント $\vec{p}$ と類似度の高いフィンガープリントの個数を Alice に教えることがプロトコルの目標となる。すると、Bob が Alice に送信する暗号文の順番をランダムに並べ変えてもプロトコルの正しさは損なわれない。そうすることで、暗号文と Bob のフィンガープリントとの対応関係が隠されていない状況と比べて、（特に Alice が繰り返し検索を行う状況を考えると）漏れる情報量がより小さくなっていると考えられる。

さらに前述の二つの対策を組み合わせることもでき、その場合、Tversky index が閾値以上となる Bob のフィンガープリントの個数は $N_{\mathrm{p,all}} - N_{\mathrm{p,dummy}}$ で与えられる。

4　安全性評価

本章では、閾値型 Tversky index の利用により Tversky index の値を隠す提案プロトコルの工夫によって、Alice に渡るフィンガープリント $\vec{q}$ の情報をどのくらい減らせているかを考察する。以下、簡略化のため、Bob のフィンガープリント $\vec{q}$ が一つだけの場合を考える。

理想的な状況での安全性　まず、提案方式の工夫が理想的に作用した状況、即ち、1 回のプロトコルで $\vec{q}$ について Alice が得る情報が $\mathsf{TI}(\vec{p}, \vec{q}) \geq \theta$ であるか否かに限られている状況を仮定する。この場合、Alice が得る情報は二択のうちの一つなので、その情報量は高々 $H(1/2, 1/2) = 1$ である（$H(\cdot)$ は（2 を底とする）シャノンエントロピー）。実際には $\mathsf{TI}(\vec{p}, \vec{q}) \geq \theta$ の場合とそうでない場合が均等に生じるとは限らないため、Alice が得る情報量はより小さいと考えられる。

一方、既存方式のように Tversky index の値自体が Alice に知られる状況を考える。Bob のフィンガープリント $\vec{q}$ がある確率分布に従うとき、Alice のフィンガープリント $\vec{p}$ を指定した状

況での $\mathsf{TI}(\vec{p},\vec{q})$ の確率分布を $T_{\vec{p}}$ で表すと、1 回のプロトコルで Alice に漏れる情報量は $H(T_{\vec{p}})$ となる。よって、Tversky index の大小以外に漏れる余分な情報量は $H(T_{\vec{p}})-1$ 以上となる。

具体例として、パラメータ $\alpha=\beta=1$ を選び、$\vec{q}$ が $\{0,1\}^\ell$ 上の一様分布に従うと仮定したとき、$\vec{p}$ の全ビットを 1 としたプロトコルで Alice に漏れる情報量 $H(T_{\vec{p}})$ を計算機で求めた結果を表 2 に示した。例えば、実際の化合物データベースにも用いられているフィンガープリントの設計方式である Maccs Keys（例えば [2] を参照）では $\ell=166$ という値を用いているが、このとき $H(T_{\vec{p}})=4.734$ となっており、Alice に漏れる余分な情報量は少なくとも 3.734 である。逆に言うと、提案方式を用いることで（理想的な状況であれば）Alice に漏れる情報量をそれだけ削減できることになる[7]。

表 2: Tversky index が漏らすフィンガープリントの情報量

ℓ	$H(T_{\vec{p}})$	ℓ	$H(T_{\vec{p}})$
10	2.706	166	4.734
20	3.207	200	4.869
50	3.868	1000	6.029
100	4.369	—	—

（$\alpha=\beta=1$、$\vec{p}$ は全ビット 1、$\vec{q}$ は一様分布に従うと仮定）

なお、仮に提案方式が理想的に働いたとしても、プロトコルの目的上、Alice に 1 ビット程度の情報はどうしても漏れてしまう。そのため、現実には同一のクライアントによる検索回数に上限を設けるなど運用による対処が必要となると考えられる。その場合でも、もし合計で同じ量の情報の漏れを許容するならば、既存方式に比べて提案方式の方がより回数の設定にゆとりがあることになる。

ダミー値を利用した場合　前項より、もし提案方式における工夫が理想的に作用していれば、Alice に漏れる余分な情報量を確かに削減できる。そこで、現実の提案方式が理想的な状況にどの程度近いかを考察する。ここでは、3.3 節での改良のうちダミー値の利用を行った場合を取り扱う。簡略化のため、残りの改良（アフィン変換とランダム置換）については考慮しない。

具体的には、Alice が閾値型 Tversky index $\overline{\mathsf{TI}}(\vec{p},\vec{q})$ とそのダミー値（およびダミー値中の「正の値」の個数）を受け取った状況で、$\overline{\mathsf{TI}}(\vec{p},\vec{q})$ の値を推定するゲームを考える。前述の「理想的な状況」は、この推定成功確率がほぼ 0 となる状況[8]と考えられる。実験の簡略化のため、閾値型 Tversky index とそのダミー値は常に「正の値」であり、k 通りの候補から一様ランダムに出現すると仮定する。その際の推定成功確率を計算機シミュレーションと少々の解析的操作によって求めたところ、例えば $k=10^3$ の場合（これは応用上も現実的な値である）、ダミー値を 10^3 個用いた場合の推定成功確率は約 4.7×10^{-3} となり、ダミー値を 10^6 個用いた場合には推定成功確率が $1/k$ に近く（約 1.3×10^{-3}）なった（シミュレーション回数は各々10^6 回である）。よって、少なくともダミー値を充分な数だけ用意すれば、Alice に漏れる余分な情報量をほぼ 0 にできることが定量的にも確かめられた。

5　形式逸脱攻撃者への対応

3.1 節で提案したプロトコルでは、Alice について Bob の情報を抜き出そうとするもののプロトコル自体は遵守するという honest-but-curious な攻撃者モデルを暗に想定していた。一方、Alice が malicious な攻撃者の場合、平文 p_i として本来想定されている 0 や 1 以外の異常値を選ぶことで Bob のフィンガープリント $\vec{q}$ の情報をより多く引き出そうとする可能性がある。例えば、p_i たちのうち一つだけを極端に大きな値とし、他を全て 0 に設定すると、Alice の受信する暗号文 c_{TI} に対応する平文が「正の値」となることと $q_i=1$ とが同値になるため、Bob のフィンガープリントの任意のビットを確実に知

[7] これは既存方式に対して、最適とは限らないある特定の攻撃戦略を仮定した場合の値なので、より優れた戦略が存在するであろうことを鑑みると、実際には情報量の削減効果はより大きいと考えられる。

[8] より正確には、（値の既知の分布以外に）何のヒントもなく値を推定する際の推定成功確率と一致する状況

ることができてしまう。

上記の攻撃を防止するためには、Alice の平文が間違いなく 0 または 1 のいずれかであることを、Bob が受信した暗号文から確かめられるようにすればよい。それには、2.3 節で言及した坂井らによる非対話ゼロ知識証明 [7] が利用できる。なお、この非対話ゼロ知識証明を適用できる加法準同型暗号方式は今のところ ElGamal 暗号 [3] に限られるため、この対策を適用する場合は上記のプロトコルにも ElGamal 暗号を用いる必要があることを注意しておく。

6 まとめ

本稿では、クライアント側とデータベース側が指定した化合物の符号化情報（フィンガープリント）の類似度を、互いの情報をできるだけ漏らさないようにしつつクライアント側に教えるプロトコルを提案し、その安全性評価を行った。提案プロトコルでは、加法準同型暗号方式を適切に利用することで、一般的な多者計算プロトコルと比べて小さい通信量・計算量および 1 往復の通信ラウンド数を達成している。より詳細な実行速度や安全性の評価については、現在準備中の full version を参照されたい。

謝辞 本研究に際して、産業技術総合研究所の松田隆弘氏と Jacob C. N. Schuldt 氏より有益なコメントを頂いたのでここで感謝する。

参考文献

[1] C. Blundo, E. De Cristofaro and P. Gasti, EsPRESSo: Efficient privacy-preserving evaluation of sample set similarity, Preprint, 2011, `http://arxiv.org/abs/1111.5062`

[2] S. K. Dogra, Script for getting MACCS keys, in: QSARWorld – free online resource for QSAR modeling, `http://www.qsarworld.com/virtual-workshop.php`

[3] T. Elgamal, A public key cryptosystem and a signature scheme based on Discrete Logarithms, *IEEE Transactions on Information Theory*, vol.IT-31, no.4, 1985, pp.469–472.

[4] C. Gentry, Fully homomorphic encryption using ideal lattices, in: *Proceedings of STOC 2009*, 2009, pp.169–178.

[5] Y. C. Martin, J. L. Kofron and L. M. Traphagen, Do structurally similar molecules have similar biological activity? *Journal of Medicinal Chemistry*, vol.45, no.19, 2002, pp.4350–4358.

[6] P. Paillier, Public-key cryptosystems based on composite degree residuosity classes, in: *Proceedings of EUROCRYPT 1999*, LNCS 1592, 1999, pp.223–238.

[7] Y. Sakai, K. Emura, G. Hanaoka, Y. Kawai and K. Omote, Towards restricting plaintext space in public key encryption, in: *Proceedings of IWSEC 2011*, LNCS 7038, 2011, pp.193–209.

[8] M. D. Singh, P. R. Krishna and A. Saxena, A privacy preserving Jaccard similarity function for mining encrypted data, in: *Proceedings of TENCON 2009 – 2009 IEEE Region 10 Conference*, 2009, `http://dx.doi.org/10.1109/TENCON.2009.5395869`

[9] A. Tversky, Features of similarity, *Psychological Review*, vol.84, 1977, pp.327–352.

[10] B. Zhang and F. Zhang, Secure similarity coefficients computation with malicious adversaries, Preprint, 2012, `http://eprint.iacr.org/2012/202`