まぜるな危険準同型暗号

江村 恵太[1]　林 卓也[1]　國廣 昇[2,4]　佐久間 淳[3,4]

概要：準同型暗号では, 暗号文を復号することなしに平文に対し演算を行うことが可能であり, クラウドサーバへの安全な計算委託などの応用が期待されている. ただし暗号化により平文に関する情報をクラウドサーバに知られない一方で, 平文の内容ごとに演算を行う/行わないを切り分けることはできない. そのため暗号化されたカルテから病気に関する統計情報を計算したい場合などにおいて, 統計結果に演算対象の病気に関するカルテとは別の病気に関するカルテの内容が混在してしまう懸念がある. 本論文では, このような誤った準同型演算処理を防止するため, 同じキーワードに関連した暗号文に対してのみ準同型演算を許す “まぜるな危険準同型暗号” を提案する. さらに準同型演算をクラウドサーバに委託する場合を考慮し, キーワードに対する安全な検索機能も実現する. 最後に提案方式を用いた内積計算の実装結果を示す.

キーワード：まぜるな危険, 準同型暗号, プライバシー保護データマイニング

Mis-operation Resistant Homomorphic Encryption

Keita Emura[1]　Takuya Hayashi[1]　Noboru Kunihiro[2,4]　Jun Sakuma[3,4]

Abstract: Let us consider a data (e.g., a medical record) is related to a keyword (e.g., a disease name), and is encrypted by using a homomorphic encryption scheme, and let a server be delegated to compute a ciphertext of a statistical value of data. Then, the server should NOT be allowed to perform the homomorphic operation against ciphertexts associated with different keywords. If such a mis-operation happens, then medical records of different diseases are unexpectedly mixed. In this paper, we propose mis-operation resistant homomorphic encryption, where no homomorphic operation is allowed against ciphertexts associated with different keywords. We also consider secure keyword search since keywords might cause identification of the patient, e.g., when the keyword indicates rare diseases. Finally, we give our implementation result of inner products of encrypted vectors.

Keywords: Mis-operation Resistance, Homomorphic Encryption, Privacy Preserving Data Mining

1. はじめに

準同型暗号を利用することで, 暗号文を復号することなく平文に関する演算を行うことができ, サーバに安全に計算を委託することができる. Gentry [12] による完全準同型暗号の提案などの準同型暗号の発展に伴い, GWAS (genome-wide association study) [19] や化合物検索 [20] などの応用が提案されている.

その有用な機能性の一方で, 準同型暗号の安全性に関しては通常の暗号と比較して弱い安全性しか達成できないことが知られている. つまり (公開鍵にアクセス可能な) 誰もが自由に準同型演算を実施可能であることは誰もが暗号文を偽造可能ということに等しく, 暗号文から平文に関する一切の情報を漏らさず, かつ一切の暗号文の改変を許さない安全性 (適応的選択暗号文攻撃に対する識別不可能性, 以下 CCA 安全性) を達成することはできない. そこで準同型暗号のモデルを修正することにより, 準同型性と CCA 安全性とを同時に達成する “鍵付き準同型暗号 (KH-PKE:

[1] 情報通信研究機構
National Institute of Information and Communications Technology (NICT)
[2] 東京大学
The University of Tokyo
[3] 筑波大学
University of Tsukuba
[4] JST CREST

Keyed Homomorphic Public Key Encryption)” が提案されている [9], [10]. 鍵付き準同型暗号では, 通常の公開鍵と復号鍵に加え準同型演算鍵が定義され, この準同型演算鍵を持つユーザに対しては準同型演算を許す一方で, 準同型演算鍵を持たないユーザに対しては CCA 安全性を保証する. これまで公開検証可能な方式 [15], [18], 鍵付き完全準同型暗号 [17], 鍵付き準同型 ID ベース暗号 [9] などが提案されている.

1.1 研究対象: まぜるな危険準同型暗号

鍵付き準同型暗号を利用することにより, “誰が準同型演算を行うことができるか” を制御することはできるが, 準同型性の制御に関しては議論の余地がある. 例えばデータ保持者 (病院など) がキーワード (病名など) に関連したデータ (カルテなど) を暗号化し, その暗号文に対する準同型演算をサーバに委託する場合を考える. なおキーワードがサーバに知られた場合に個人が特定される場合 (非常に稀な病気の場合など) を考慮し, サーバに対してもキーワードは秘匿されると想定する. 受信者があるクエリ (例えば癌患者の平均体重) をサーバに送付すると, サーバは準同型演算を行い, その結果の暗号文を受信者に送付, 受信者はその暗号文を復号することで癌患者の平均体重を得ることができる. この例において, 異なる病名に関連付いた暗号文に対する準同型演算をサーバに許可するべきではない. しかしながら現状の準同型暗号や鍵付き準同型暗号では, このような誤った準同型演算を防ぐ手立てがなく, 復号時に初めて検出されるか, もしくは統計値からは検出できない懸念がある. そこで異なるキーワードに関連付いた暗号文に対し準同型演算を行おうとした場合に警告を発する “まぜるな危険 (Mis-operation Resistance)” 機能が望まれる.

まぜるな危険機能を達成する方策として, タグベース暗号 [16] のようにある “タグ” を暗号文に付加することを考える. 暗号文からは平文に関する情報が漏れないことから, タグを用いることは非常に有用であり, 暗号文がキーワードに関連付いていることから, キーワードをタグとみなすことが自然であるように思われる. しかしながら, 前述した通りキーワードも秘匿すべき情報である. そこで, 以下では検索可能暗号の枠組みを導入することを考える.

1.2 安易な構成とその限界

最も簡単な構成として, 準同型暗号と検索可能暗号 (public key encryption with keyword search (PEKS) [5]) とを組み合わせることが挙げられる. 病院はカルテを準同型暗号方式で暗号化, 病名を PEKS 方式で暗号化する. ある病名に関するカルテの統計情報を取得する場合, 病名を検索可能なトラップドアをサーバに送付, サーバは検索を行い対象の PEKS 暗号文を見つけ, 同時に保存されている準同型暗号方式の暗号文に対し準同型演算を行う. しかしながら, この構成では暗号文の部分的な置き換えが可能であり, CCA 安全性を達成できない. また準同型暗号方式の暗号文は任意に改変可能であり, いくらサーバがプロトコルに従うと仮定したとしても, 現実的には誤った準同型演算が行われる可能性がある. 検索機能とデータの暗号化を同時に扱いつつ, CCA 安全性を達成する方式として, PEKS/PKE [3], [21] が提案されているが, 公開鍵暗号部分の準同型性を保存しないため, まぜるな危険機能を達成する方策としては適用できない.

次に二重暗号化により, まぜるな危険機能の実現を試みる. 準同型暗号方式に加え, キーワードごとに公開鍵と秘密鍵とのペアを作成し, データがあるキーワードに関連している場合にはデータを準同型暗号方式で暗号化, その暗号文を対象となるキーワード用の公開鍵でさらに暗号化する. サーバがあるキーワードに関連付けされた暗号文に対する準同型演算を行う場合, まず対象の秘密鍵で暗号文を復号し, 得られた準同型暗号方式の暗号文に対し, 準同型演算を行う. しかしながら, この二重暗号化方式には様々なデメリットがある. まずキーワードの個数分の鍵ペアを管理する必要がある. また, サーバが複数のキーワード用の秘密鍵を管理する場合, キーワードによらず準同型暗号方式の暗号文を得られてしまうことから, 結局異なるキーワードに関連付いた暗号文に対し準同型演算を行うことが可能となってしまう.

識別不能難読化 [4] や関数型暗号 [6] を用いると, まぜるな危険機能が実現できるかも知れない (例えば平文空間を適切に分割し, 平文が同じ空間に属している場合のみに準同型演算を許すようなプログラム/関数を定義するなど). しかしながら, その効率は現状実用とは程遠く, そのため本論文では考慮しない. 同様に, 準同型演算として関数族 $\mathcal{F}$ に属する関数のみを許す Targeted Malleability [6] において準同型演算可能な平文空間を適切に定義することで, まぜるな危険機能が実現できるかも知れない. しかしながら, 準同型演算時にある関数 $f \in \mathcal{F}$ を適用したことを証明する非対話ゼロ知識証明を付加する必要があり, まぜるな危険機能を実現する言語に対する具体的な構成は非自明である.

最後に鍵付き準同型 ID ベース暗号 [9] の利用を試みる. 鍵付き ID ベース暗号では, 準同型演算鍵が ID ごとに定義され, 同じ ID で暗号化された暗号文に対して準同型演算を行うことができる.[*1] また, 論文 [9] では Gentry ID ベース暗号 [11] をベースに構成されており, Gentry ID ベース暗号が匿名性 (暗号文から ID が漏洩しない) を持つこと, かつ匿名 ID ベース暗号からは PEKS が一般的に構成可能 (Abdalla ら [2]) であることから, この Gentry ベース

[*1] Kiltz タグベース暗号 [16] や Gentry-Sahai-Waters ID ベース完全準同型暗号 [13] も同様に, 同じタグ/ID で暗号化された暗号文に対して準同型演算を行うことができる. しかしながら検索機能や鍵付き準同型性をみたさないことから, 本論文では考慮しない.

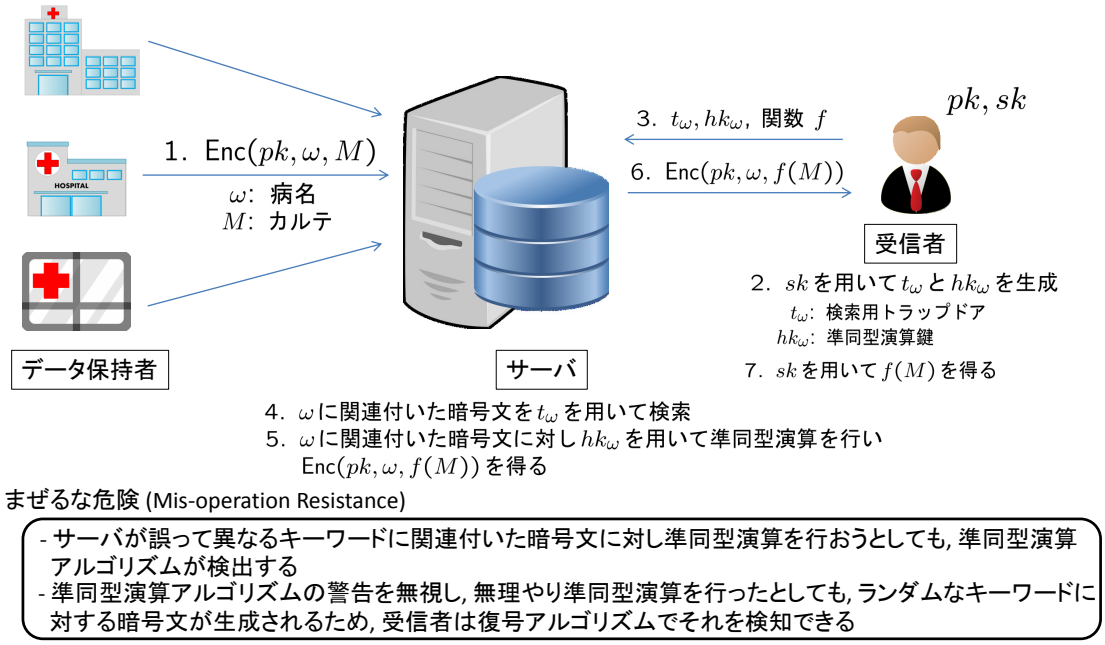

図 1 まぜるな危険準同型暗号 (MR-SHE) のフレームワーク

鍵付き ID ベース暗号に Abdalla らの構成を利用することで, 目的の機能が達成されると期待できる. しかしながら, Abdalla らの構成では, データの暗号化を考慮していない. より具体的には, ランダムに平文 R を選び, キーワードを ID として匿名 ID ベース暗号方式で暗号化し, R と ID ベース暗号方式の暗号文を検索可能暗号方式の暗号文, この ID に対する秘密鍵を検索用トラップドアとする. 検索時には, ID ベース暗号方式の暗号文をトラップドアで復号し, 復号結果が R となるかどうかを確認する. 検索のために平文そのものを暗号文に含める必要があり, すなわち病名を隠すためにカルテをサーバに与える必要があるため, 単純に Abdalla らの構成を適用するだけでは, まぜるな危険機能を達成することはできない.

1.3 本論文の貢献

本論文では, まぜるな危険機能を持つ準同型暗号を提案する. 以下, Mis-operation Resistant Searchable Homomorphic Encryption (MR-SHE) と呼び, 以下の機能をサポートする.

データ秘匿性: キーワードとデータとが同時に暗号化され, 暗号文からはキーワードとデータに関する情報が漏洩しない (PEKS/PKE と同様).

検索可能性: 安全なキーワード検索を実現 (PEKS と同様).

鍵付き準同型性: 準同型演算鍵を持たないユーザに対して準同型演算を許さず, かつ CCA 安全性を保証する (鍵付き準同型暗号と同様).

まぜるな危険性: 異なるキーワードに関連した暗号文に対し準同型演算を行おうとした際に準同型演算アルゴリズムが警告を発する. 警告を無視し無理やり準同型演算を行ったとしても, 復号時に復号アルゴリズムが棄却することで, そのような準同型演算が行われたことを検知できる.

図 1 に MR-SHE のフレームワークを示す. 受信者 (研究者) が鍵ペア (pk, sk) を生成し, データ保持者 (病院) はカルテ M と病名 ω を pk を用いて暗号化し, その暗号文をサーバに送付する.[*2] 暗号文からは M と ω に関する情報は漏えいしない. 病名 ω に対し, 受信者は検索用トラップドア t_ω を sk を用いて生成する. また, 病名 ω に関連付いた暗号文用の準同型演算鍵 hk_ω を sk から生成する. 受信者はセキュアチャネルを通じて t_ω と hk_ω をサーバに送付する. サーバは t_ω を用いて ω に関連付いた暗号文を検索し, hk_ω を用いて検索した暗号文に対する準同型演算 f を行う. 最終的にサーバは $f(M)$ の暗号文を計算し, 受信者に送付, 受信者は sk でその暗号文を復号することで $f(M)$ を得る. なお提案方式では, sk をマスター鍵としてキーワード ω に関連付いた暗号文用の復号鍵を生成し, その鍵を用いて復号を行う. まぜるな危険機能により, もしサーバが ω' に関連付いた暗号文に対し hk_ω $(\omega \neq \omega')$ を用いて準同型演算を行ったとしても, 準同型演算アルゴリズムがこれを検出し警告を発することができる. 加えて, 準同型演算アルゴリズムの警告を無視し, サーバが無理やり準同型演算を行ったとしても, ランダムなキーワードに対する暗号文が生成される. そのため受信者は復号アルゴリズムによりそのような不正な操作が行われたことを検知することが

[*2] 受信者がこの暗号文を復号可能なことが望ましくない場合, データ保持者とサーバ間にセキュアチャネルを仮定すればよい.

できる. 復号アルゴリズムによる暗号文の棄却についての詳細は**注意**を参照されたい.

提案 MR-SHE 方式では, 鍵付き準同型 ID ベース暗号 [9] における準同型演算鍵を検索用トラップドアとして使用し (つまり $t_\omega = hk_\omega$), 準同型演算アルゴリズム内で実行していた暗号文の正当性確認処理を検索アルゴリズムとして利用する. このとき, Abdalla らの構成とは異なり平文を暗号文に含める必要はない.

提案 MR-SHE 方式は (双線型群のターゲット群上の) 乗法準同型性をみたす. なお Lifted ElGamal 暗号同様に, 平文空間が小さい整数であれば加法準同型をサポートできる [8]. そこで片方のベクトルのみが暗号化されている場合のベクトル内積計算を, 加法準同型演算を用いて実装する. さらに両方のベクトルが暗号化されている場合のベクトル内積計算も Catalano-Fiore 変換 [7] を利用することで実装する. 実装には PBC ライブラリ [1] を使用した. 内積値を計算することにより, 例えば暗号化したまま χ^2 検定を行うことが可能となる.

注意: 受信者はマスター鍵 sk を所持しているため, 受信者が暗号文からキーワードを得られる可能性があり, ランダムなキーワードに対する暗号文であっても復号が可能である可能性は (証明可能性の意味で) 捨てきれない. このような状況を防ぐためには, マスター鍵を所持していても匿名性を達成する ID ベース暗号 [14] を応用することも考えられる. しかしながら, 受信者の目的はある ω に関連付いたデータに対する準同型演算結果を取得することであり, そのような無意味なキーワードに関連付いた暗号文の復号を行うことは考えにくいことから, 本論文では問題視しない.

1.4 暗号化データに対する χ^2 独立性検定

以下, まぜるな危険機能が有効なシナリオを示す. 研究者 ((pk, sk) を所持していると仮定) と病院 (病院 1 と病院 2) が SNP (一塩基多形, Single Nucleotide Polymorphism) 間の独立性検定を行うため, 分割表 (表 1) を計算する場合を考える. ここで Case 群は検定対象とする病気を罹患している患者群, Control 群はそうでない患者群である.

表 1 分割表

	遺伝的特徴 A を持つ	遺伝的特徴 A を持たない	
Case	n_{1A}	n_{1a}	$n_1 := n_{1A} + n_{1a}$
Control	n_{2A}	n_{2a}	$n_2 := n_{2A} + n_{2a}$
	$n_A := n_{1A} + n_{2A}$	$n_a := n_{1a} + n_{2a}$	n

ある 2 対立遺伝子 A と a を考える. n を全体の患者数とし, n_{1A} を遺伝的特徴 A を持ち, かつ病気を罹患している人数, 以下同様とする. 病院 1 がベクトル $\boldsymbol{x}_0 := (x_{0,1}, \ldots, x_{0,\ell}) \in \mathbb{Z}_2^\ell$ (もし患者 i が病気を罹患している場合 $x_{0,i} = 1$, そうでない場合 $x_{0,i} = 0$) を持ち, 病院 2 がベクトル $\boldsymbol{x}_1 := (x_{1,1}, \ldots, x_{1,\ell}) \in \mathbb{Z}_2^\ell$ (もし患者 i が遺伝的特徴 A を持つ場合 $x_{1,i} = 1$, そうでない場合 $x_{1,i} = 0$) を持つと仮定する. このとき,

$$n_{1A} = \sum_{i=1}^{\ell} x_{0,i} x_{1,i}$$

と計算できる. n, n_1, n_2, n_A, n_a は分割表に含まれているため, 研究者及び病院には公開の値であるとする. すなわち, n_{1A} が計算できれば他の値も計算可能となる. 分割表より χ^2 検定の統計検定量は.

$$T = \frac{n(n_{1A}n_{2a} - n_{1a}n_{2A})^2}{n_A n_a n_1 n_2}$$

で計算される. より大きい T は SNP と病気との関連が強いことを示す. 以下, n_{1A} の計算を必ずしも信頼できないサーバに委託することを考える. $\boldsymbol{x}_0$ 及び $\boldsymbol{x}_1$ に加え, どの病気に関する検定であるかもサーバに対しては秘匿すべき情報である. そこで病院 1, 2 はそれぞれ $\boldsymbol{x}_0$ と $\boldsymbol{x}_1$ を病名をキーワードとして暗号化し, その暗号文をサーバに送付する. 研究者が検定対象の病名に対する検索用トラップドアを計算し, サーバに送付, サーバは検索を行い, 準同型演算を行うことで n_{1A} の暗号文を計算し研究者に送付, 研究者は復号処理を行い n_{1A} を得る. なおサーバには別の病気に対する計算も委託している場合において, 異なる病気に対する値が誤って混在するようなことがあってはならない. まぜるな危険機能により, そのような可能性を排除できる.

2. MR-SHE の定義

MR-SHE 方式 $\mathcal{MR}\text{-}\mathcal{SHE}$ は (MR-SHE.KeyGen, MR-SHE.HomKeyGen, MR-SHE.Trapdoor, MR-SHE.Enc, MR-SHE.Test, MR-SHE.Dec, MR-SHE.Eval) から成る. 以下 $\kappa \in \mathbb{N}$ をセキュリティパラメータとする.

MR-SHE.KeyGen: 鍵生成アルゴリズムは 1^κ を入力とし, 受信者の公開鍵 pk と秘密鍵 sk を出力する. なおキーワード空間 $\mathcal{W}$ 及び平文空間 $\mathcal{M}$ は pk に含まれていると仮定する.

MR-SHE.HomKeyGen: 準同型演算鍵生成アルゴリズムは pk, sk, キーワード $\omega \in \mathcal{W}$ を入力とし, 準同型演算鍵 hk_ω を出力する.

MR-SHE.Trapdoor: 検索用トラップドア生成アルゴリズムは pk, sk, キーワード $\omega \in \mathcal{W}$ を入力とし, 検索用トラップドア t_ω を出力する.

MR-SHE.Enc: 暗号化アルゴリズムは pk, ω, M を入力とし, 暗号文 C を出力する.

MR-SHE.Test: テストアルゴリズムは pk, t_ω, C を入力とし, 1 または 0 を出力する.

MR-SHE.Dec: 復号アルゴリズムは pk, sk, ω, C を入力とし, M または $\perp$ を出力する.

MR-SHE.Eval: 準同型演算アルゴリズムは pk, hk_ω, 2 つの暗号文 C_1 と C_2 を入力とし, 暗号文 C または $\perp$ を出力する.

Correctness: 全ての $(pk, sk) \leftarrow \mathsf{MR\text{-}SHE.KeyGen}(1^\kappa)$, $\omega \in \mathcal{K}$, $M \in \mathcal{M}$ に対し,

- $C \leftarrow \mathsf{MR\text{-}SHE.Enc}(pk, \omega, M)$ と $t_\omega \leftarrow \mathsf{MR\text{-}SHE.Trapdoor}(pk, sk, \omega)$ に対し, $\mathsf{MR\text{-}SHE.Test}(pk, t_\omega, C) = 1$ が成り立つ
- $\mathsf{MR\text{-}SHE.Dec}(pk, sk, \omega, C) = M$ が成り立つ.

準同型性: 全ての $(pk, sk) \leftarrow \mathsf{MR\text{-}SHE.KeyGen}(1^\kappa)$, $\omega \in \mathcal{K}$, $M_1, M_2 \in \mathcal{M}$, $hk_\omega \leftarrow \mathsf{MR\text{-}SHE.HomKeyGen}(pk, sk, \omega)$ に対し,

- $C \leftarrow \mathsf{MR\text{-}SHE.Eval}(pk, hk_\omega, C_1, C_2)$ に対し, $\mathsf{MR\text{-}SHE.Dec}(pk, sk, \omega, C) = M_1 \odot M_2$ が成り立つ. ここで $\odot$ は $\mathcal{M}$ 上の二項演算であり, $C_1 \leftarrow \mathsf{MR\text{-}SHE.Enc}(pk, \omega, M_1)$, $C_2 \leftarrow \mathsf{MR\text{-}SHE.Enc}(pk, \omega, M_2)$ とする.

次に暗号化時し指定されたキーワードと異なるキーワードに対するトラップドアを入力とした際に MR-SHE.Test が 1 を出力する確率が無視できることを保証する Consistency を定義する. なお準同型演算後の暗号文も対象とした定義も可能であるが, 提案方式では, 暗号化アルゴリズムで作成された暗号文と準同型演算後の暗号文とは同じ形であるため, 準同型演算後の暗号文に対する Cnsistency はここでは定義しない.

Definition2.1 (Consistency) 任意の確率的多項式攻撃者 $\mathcal{A}$ とセキュリティパラメータ $\kappa \in \mathbb{N}$ に対し, experiment $\mathrm{Exp}^{\mathrm{consist}}_{\mathcal{MR}\text{-}\mathcal{SHE},\mathcal{A}}(\kappa)$ を以下で定義する.

$\mathrm{Exp}^{\mathrm{consist}}_{\mathcal{MR}\text{-}\mathcal{SHE},\mathcal{A}}(\kappa)$:
$(pk, sk) \leftarrow \mathsf{MR\text{-}SHE.KeyGen}(1^\kappa)$
$(\omega, \omega', M) \leftarrow \mathcal{A}(pk)$
$\omega, \omega' \in \mathcal{W};\ \omega \neq \omega';\ M \in \mathcal{M}$
$C \leftarrow \mathsf{MR\text{-}SHE.Enc}(pk, \omega, M)$
$t_{\omega'} \leftarrow \mathsf{MR\text{-}SHE.Trapdoor}(pk, sk, \omega')$
if $\mathsf{MR\text{-}SHE.Test}(pk, t_{\omega'}, C) = 1$ then return 1
else return 0

もしアドバンテージ $\mathrm{Adv}^{\mathrm{consist}}_{\mathcal{MR}\text{-}\mathcal{SHE},\mathcal{A}}(\kappa) := \Pr[\mathrm{Exp}^{\mathrm{consist}}_{\mathcal{MR}\text{-}\mathcal{SHE},\mathcal{A}}(\kappa) = 1]$ が無視できるとき, $\mathcal{MR}\text{-}\mathcal{SHE}$ は Consistency をみたすと定義する.

他にデータ秘匿性 (暗号文から平文 M の情報が漏れない) 及びキーワード秘匿性 (暗号文からキーワード ω の情報が漏れない) が定義される (詳細は Full version).

3. 提案 MR-SHE 方式

提案 MR-SHE 方式では, 鍵付き準同型 ID ベース暗号方式 [9] の準同型演算鍵を検索用トラップドアとして使用し, 検索手順と準同型演算アルゴリズム内での暗号文正当性確認手順とは同じ手続きとなる. なお元々の準同型演算アルゴリズムでは ID と暗号文との両方を入力としていた. 提案方式では, キーワード自体を知ることなく準同型演算を行う必要があるため, 鍵付き準同型 ID ベース暗号の準同型演算鍵に追加で g^ω を含めることで, $g_1^s g^{-s\omega} = (g_1^{-1} g^\omega)^{-s}$ と計算する. なお匿名性の安全性定義では攻撃者が選択したキーワード (ω_0^*, ω_1^*) に対する識別不可能性を要求しており, その攻撃者はそもそも自身で $g^{\omega_0^*}$ や $g^{\omega_1^*}$ を計算できることから, この変更が安全性に影響を与えることはない.

以下 $\mathbb{G}$, $\mathbb{G}_T$ を素数位数 p を持つ群, $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ を双線型写像, $\mathcal{W} := \mathbb{Z}_p$ をキーワード空間, $\mathcal{M} := \mathbb{G}_T$ を平文空間, $\{\Gamma = \Gamma_{hk} : \mathbb{G}^4 \to \{0, 1, \ldots, p-1\} \mid hk \in \mathcal{HK}\}$ をターゲット衝突困難ハッシュ関数族, $f : \mathbb{G}_T \to \mathcal{Y}$ をスムース関数 [9] とする[*3].

なお Gentry ID ベース暗号と同様に, あるキーワードに対しては 1 つしか鍵が生成されないとする. そのため, 復号鍵 $\{(r_{\omega,i}, h_{\omega,i})\}_{i=1,2,3,4}$ が一度生成された場合, 別の復号鍵は生成されないとし, 他のアルゴリズムでも同じ値を共有する. 以下, 煩雑さを避けるため, まだ鍵が生成されていないと仮定した場合の処理のみを記述する.

$\mathsf{MR\text{-}SHE.KeyGen}(1^\kappa)$: $hk \xleftarrow{\$} \mathcal{HK}$, $g \xleftarrow{\$} \mathbb{G}$, $h_1, h_2, h_3, h_4 \xleftarrow{\$} \mathbb{G}$, $\alpha \xleftarrow{\$} \mathbb{Z}_p$ を選択し, $g_1 \leftarrow g^\alpha$ を計算, $pk = (g, g_1, h_1, h_2, h_3, h_4, hk, f)$ 及び $sk = \alpha$ を出力する.

$\mathsf{MR\text{-}SHE.HomKeyGen}(pk, sk, \omega)$: $i = 3, 4$ に対し $r_{\omega,i} \xleftarrow{\$} \mathbb{Z}_p$ を選択し, $h_{\omega,i} \leftarrow (h_i g^{-r_{\omega,i}})^{1/(\alpha-\omega)}$ を計算, $hk_\omega = (g^\omega, (r_{\omega,3}, h_{\omega,3}), (r_{\omega,4}, h_{\omega,4}))$ を出力する.

$\mathsf{MR\text{-}SHE.Trapdoor}(pk, sk, \omega)$: $i = 3, 4$ に対し $r_{\omega,i} \xleftarrow{\$} \mathbb{Z}_p$ を選択し, $h_{\omega,i} \leftarrow (h_i g^{-r_{\omega,i}})^{1/(\alpha-\omega)}$ を計算, $hk_\omega = (g^\omega, (r_{\omega,3}, h_{\omega,3}), (r_{\omega,4}, h_{\omega,4}))$ を出力する.

$\mathsf{MR\text{-}SHE.Enc}(pk, \omega, M)$: $s \xleftarrow{\$} \mathbb{Z}_p$ を選択し, $c_1 \leftarrow g_1^s g^{-s\omega}$, $c_2 \leftarrow e(g, g)^s$, $c_3 \leftarrow M \cdot e(g, h_1)^{-s}$, $c_4 \leftarrow e(g, h_2)^s$, $\delta \leftarrow \Gamma_{hk}(c_1, c_2, c_3, c_4)$, $c_5 \leftarrow e(g, h_3)^s e(g, h_4)^{s\delta}$, $\tau \leftarrow f(c_5)$ を計算し, $C = (c_1, c_2, c_3, c_4, \tau)$ を出力する.

$\mathsf{MR\text{-}SHE.Test}(pk, t_\omega, C)$: t_ω を $(g^\omega, (r_{\omega,3}, h_{\omega,3}), (r_{\omega,4}, h_{\omega,4}))$, C を $(c_1, c_2, c_3, c_4, \tau)$ とする. $\delta \leftarrow \Gamma_{hk}(c_1, c_2, c_3, c_4)$ を計算し, $\tau = f(e(c_1, h_{\omega,3} h_{\omega,4}^\delta) c_2^{r_{\omega,3} + r_{\omega,4}\delta})$ であれば 1 を, そうでない場合 0 を出力する.

[*3] $f : \mathcal{X} \to \mathcal{Y}$ をハッシュ関数とする. もし $\mathsf{Smth}_f := \max_{y \in \mathcal{Y}} \Pr_{x \xleftarrow{\$} \mathcal{X}}[f(x) = y]$ が ϵ より大きくない場合, f は ϵ スムースであると定義する. ϵ が無視できる場合, 単に f はスムースであると定義する.

MR-SHE.Dec(pk, sk, ω, C): $sk = \alpha$, $C = (c_1, c_2, c_3, c_4, \tau)$ とする. $i = 1, 2, 3, 4$ に対し, $r_{\omega,i} \xleftarrow{\$} \mathbb{Z}_p$ を選び, $h_{\omega,i} \leftarrow (h_i g^{-r_{\omega,i}})^{1/(\alpha-\omega)}$ を計算する. $\delta \leftarrow \Gamma_{hk}(c_1, c_2, c_3, c_4)$, $c'_4 \leftarrow e(c_1, h_{\omega,2}) c_2^{r_{\omega,2}}$, $c_5 \leftarrow e(c_1, h_{\omega,3} h_{\omega,4}^{\delta}) c_2^{r_{\omega,3}+r_{\omega,4}\delta}$ を計算し, もし $c'_4 \neq c_4$ または $\tau \neq f(c_5)$ の場合, $\perp$ を出力, そうでない場合, $M \leftarrow c_3 \cdot e(c_1, h_{\omega,1}) c_2^{r_{\omega,1}}$ を出力する.

MR-SHE.Eval$(pk, hk_\omega, C_1, C_2)$: hk_ω を $(g^\omega, (r_{\omega,3}, h_{\omega,3}), (r_{\omega,4}, h_{\omega,4}))$, C_1 を $(c_{1,1}, c_{1,2}, c_{1,3}, c_{1,4}, \tau_1)$, C_2 as $(c_{2,1}, c_{2,2}, c_{2,3}, c_{2,4}, \tau_2)$ とする.

[暗号文正当性確認]: $\delta_1 \leftarrow \Gamma_{hk}(c_{1,1}, c_{1,2}, c_{1,3}, c_{1,4})$, $c_{1,5} = e(c_{1,1}, h_{\omega,3} h_{\omega,4}^{\delta_1}) c_{1,2}^{r_{\omega,3}+r_{\omega,4}\delta_1}$, $\delta_2 \leftarrow \Gamma_{hk}(c_{2,1}, c_{2,2}, c_{2,3}, c_{2,4})$, $c_{2,5} = e(c_{2,1}, h_{\omega,3} h_{\omega,4}^{\delta_2}) \cdot c_{2,2}^{r_{\omega,3}+r_{\omega,4}\delta_2}$ を計算する. もし $\tau_1 \neq f(c_{1,5})$ または $\tau_2 \neq f(c_{2,5})$ の場合, $\perp$ を出力する.

[準同型演算]: $s \xleftarrow{\$} \mathbb{Z}_p$ を選択し, $c_1 \leftarrow c_{1,1} c_{2,1} \cdot g_1^s g^{-s\omega}$, $c_2 \leftarrow c_{1,2} c_{2,2} \cdot e(g,g)^s$, $c_3 \leftarrow c_{1,3} c_{2,3} \cdot e(g, h_1)^{-s}$, $c_4 \leftarrow c_{1,4} c_{2,4} \cdot e(g, h_2)^s$, $\delta \leftarrow \Gamma_{hk}(c_1, c_2, c_3, c_4)$, $c_5 \leftarrow e(c_1, h_{\omega,3} h_{\omega,4}^{\delta}) \cdot c_2^{r_{\omega,3}+r_{\omega,4}\delta}$, $\tau \leftarrow f(c_5)$ を計算, $C = (c_1, c_2, c_3, c_4, \tau)$ を出力する.

データ秘匿性及びキーワード秘匿性は, それぞれ KH-IBE 方式の安全性に帰着される. 以下, 離散対数仮定及び f がスムース関数であるという仮定の下, 提案方式が Consistency をみたすことを示す. もし $\omega = \alpha$ または $\omega' = \alpha$ の場合, 離散対数問題 $(g, g_1 := g^\alpha)$ を解くアルゴリズムを容易に構成できる. 以下, $\omega, \omega' \neq \alpha$ と仮定する. $C = (c_1, c_2, c_3, c_4, \tau)$ を experiment で生成された暗号文とし, $\delta \leftarrow \Gamma_{hk}(c_1, c_2, c_3, c_4)$ に対し, $c_5 \leftarrow e(g, h_3)^s e(g, h_4)^{s\delta}$ 及び $\tau = f(c_5)$ とする. ここで s は (c_1, c_2, c_3, c_4) の計算に用いられた乱数である. $c'_5 := e(c_1, h_{\omega',3} h_{\omega',4}^{\delta}) c_2^{r_{\omega',3}+r_{\omega',4}\delta}$ とおく. MR-SHE.Test$(pk, t_{\omega'}, C) = 1$ が成り立つことから, (1) $c_5 = c'_5$ または (2) $c_5 \neq c'_5$ かつ $\tau = f(c'_5)$ が成り立つ. f がスムース関数であることから, Case (2) が起こる確率は無視できる. 以下, Case (1) が起こる確率を評価する.

Case (1):

$$e(c_1, h_{\omega',3} h_{\omega',4}^{\delta}) c_2^{r_{\omega',3}+r_{\omega',4}\delta}$$
$$= e(g, h_3)^{\frac{\alpha-\omega}{\alpha-\omega'} s} e(g, h_4)^{\frac{\alpha-\omega}{\alpha-\omega'} s\delta} e(g,g)^{s(r_{\omega',3}+\delta r_{\omega',4})(1-\frac{\alpha-\omega}{\alpha-\omega'})}$$

が成り立つ. ここで,

$$c_5 = e(g, h_3)^s e(g, h_4)^{s\delta}$$
$$= e(g, h_3)^{\frac{\alpha-\omega}{\alpha-\omega'} s} e(g, h_4)^{\frac{\alpha-\omega}{\alpha-\omega'} s\delta} e(g,g)^{s(r_{\omega',3}+\delta r_{\omega',4})(1-\frac{\alpha-\omega}{\alpha-\omega'})}$$

が成り立つ. $A := \frac{\alpha-\omega}{\alpha-\omega'}$ とおくと

$$s \log_g h_3 + s\delta \log_g h_4$$
$$= As \log_g h_3 + As\delta \log_g h_4 + s(r_{\omega',3} + \delta r_{\omega',4})(1 - A)$$

が成り立つ. ここから

$$s(1-A)(\log_g h_3 + \delta \log_g h_4 - r_{\omega',3} - \delta r_{\omega',4}) = 0$$

を得る. $\omega \neq \omega'$ であるため, $1 - A = 1 - \frac{\alpha-\omega}{\alpha-\omega'} \neq 0$ である. ここで ($s \xleftarrow{\$} \mathbb{Z}_p$ であるため) $s \neq 0$ と仮定する. $\log_g h_3 + \delta \log_g h_4 - r_{\omega',3} - \delta r_{\omega',4} = 0$ が成り立つため, 離散対数問題 (g, h_3) を解くアルゴリズムを容易に構成できる. 以上より, Case (1) が起こる確率は無視できる.

4. 実装結果

本章では, 提案 MR-SHE 方式を利用した内積計算の実装結果を示す.

4.1 アルゴリズム

本実装では PBC ライブラリ [1] (pbc-0.5.14, 楕円曲線 $y^2 = x^3 + x$ (Type A), 定義体は標数 512bit の素体) を使用した. 環境は CPU: Xeon E5-2660 v3 @ 2.60GHz, gcc 4.9.2, openssl 1.0.2d である. SHA512 を Γ 及び f として使用し, 全ての事前計算可能なペアリングの値, $e(g, h_i)$ $(i = 1, 2, 3, 4)$, は pk に含まれていると仮定する. まず各アルゴリズムのベンチマークを表 2 に示す. 全てのアルゴリズムが数 msec 程度で実行可能である.

表 2 各アルゴリズムの計算時間

アルゴリズム	計算時間 (msec)
MR-SHE.KeyGen	9.5
MR-SHE.HomKeyGen	1.0
MR-SHE.Trapdoor	1.0
MR-SHE.Enc	0.5
MR-SHE.Dec	4.7
MR-SHE.Test	2.2
MR-SHE.Eval	8.1

4.2 複数暗号文に対する一括準同型演算

MR-SHE.Eval アルゴリズムでは, 2 つの暗号文 C_1, C_2 を入力としてとる. 本節では, 2 つ以上の暗号文 $(C_1, C_2, \ldots, C_L)$ (L はセキュリティパラメータの多項式) に対し一括で準同型演算を行うアルゴリズム MR-SHE.mEval を提案する. なお下記のように演算しても安全性に影響を与えない. 詳細は Full version を参照されたい.

MR-SHE.mEval$(pk, hk_\omega, \{C_i\}_{i=1}^L)$: hk_ω を $(g^\omega, (r_{\omega,3}, h_{\omega,3}), (r_{\omega,4}, h_{\omega,4}))$, C_i を $(c_{i,1}, c_{i,2}, c_{i,3}, c_{i,4}, \tau_i)$ $(i = 1, \ldots, L)$ とする.

[暗号文正当性確認]: もし $\delta_i \leftarrow \Gamma_{hk}(c_{i,1}, c_{i,2}, c_{i,3}, c_{i,4})$, $c_{i,5} = e(c_{i,1}, h_{\omega,3} h_{\omega,4}^{\delta_1}) \cdot c_{i,2}^{r_{\omega,3}+r_{\omega,4}\delta_i}$ に対し, $\tau_i \neq f(c_{i,5})$ をみたす $i \in [1, L]$ が存在した場合, $\perp$ を出力する.

[準同型演算]: $s \xleftarrow{\$} \mathbb{Z}_p$ を選び, 以下を計算する.

$$c_1 \leftarrow g_1^s g^{-s\omega} \prod_{i=1}^{L} c_{i,1}, \; c_2 \leftarrow e(g,g)^s \prod_{i=1}^{L} c_{i,2}$$

$$c_3 \leftarrow e(g,h_1)^{-s}\prod_{i=1}^{L} c_{i,3},\ c_4 \leftarrow e(g,h_2)^{s}\prod_{i=1}^{L} c_{i,4}$$
$$\delta \leftarrow \Gamma_{hk}(c_1,c_2,c_3,c_4)$$
$$c_5 \leftarrow e(c_1, h_{\omega,3}h_{\omega,4}^{\delta})c_2^{r_{\omega,3}+r_{\omega,4}\delta}$$
$$\tau \leftarrow f(c_5)$$

$C=(c_1,c_2,c_3,c_4,\tau)$ を出力する.

以下, 図 2 に MR-SHE.mEval アルゴリズムのベンチマークを示す. $L=4,8,16,\ldots,8192\ (=2^{13})$ を入力する暗号文数とし, 横軸に $\log_2(L)$ の値を示す. 暗号文数が増えるに従い, MR-SHE.Eval アルゴリズムを $L-1$ 回実行した場合と比較して約 7 倍高速に準同型演算を行うことができる.

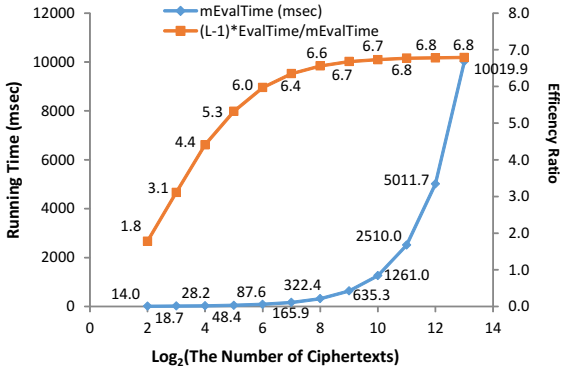

図 2　MR-SHE.mEval の計算時間

4.3　暗号文ベクトルに対する内積演算

本章では, ℓ 次元バイナリベクトル $\boldsymbol{x}_0 := (x_{0,1},\ldots,x_{0,\ell}) \in \mathbb{Z}_2^{\ell}$, $\boldsymbol{x}_1 := (x_{1,1},\ldots,x_{1,\ell}) \in \mathbb{Z}_2^{\ell}$ の内積 $\sum_{j=1}^{\ell} x_{0,j}x_{1,j}$ に対する暗号文計算及びその復号の実装結果を示す. 本論文では扱わないが, 入力を小さな整数のベクトルに容易に拡張できる. 詳細は Full version を参照されたい.

4.3.1　暗号文ベクトル同士の内積

$\boldsymbol{x}_0, \boldsymbol{x}_1$ 共に暗号化されている場合, 加法に加えて 1 回の乗法演算を必要とする. 加法準同型は Lifted ElGamal 手法を用い, 乗法演算については Catalano-Fiore 変換 [7] を利用する. 実装結果を図 3 に示す. $\ell=4,8,16,\ldots,8192=2^{13}$ をベクトルの次元とし, 横軸に $\log_2(\ell)$ の値を示す. IP Time は内積値の暗号文の計算時間, Dec after IP はその暗号文の復号時間を示す. また Del ratio は全体の計算時間のうち, どの程度をサーバに委託できたか (=IP Time/(IP Time+Dec after IP)) を示す.

比較的高次元のベクトルに対しても, 準同型演算が 36 sec, 復号に 70 sec 程度あれば実行可能である. なお Catalano-Fiore 変換のため, 最終的な復号回数がベクトルの次元 ℓ に依存することから, サーバへの計算委託比率が次元の増大

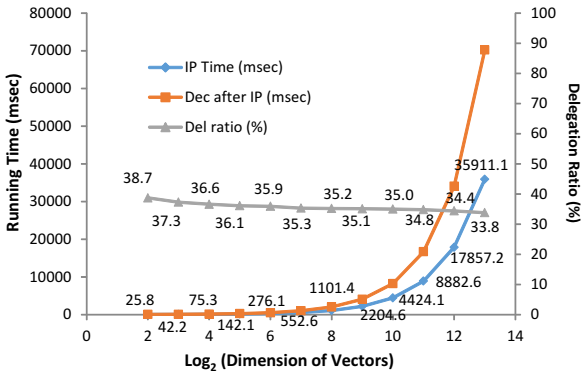

図 3　暗号文ベクトル同士の内積の計算時間

に伴って悪化する. さらに厳密な意味では CCA 安全性を満足しない. これらを避けるために完全準同型性をサポートする MR-SHE 方式の提案が望まれる.

4.3.2　暗号文ベクトルと平文ベクトルの内積

$\boldsymbol{x}_1$ のみが暗号化されている場合, 加法準同型性のみで内積の暗号文を計算することが可能である. この場合, データ $\boldsymbol{x}_0$ を所持するユーザがサーバの役割を担うこととなる (論文 [20] と同様のセッティング). 実装結果を図 4 に示す. Catalano-Fiore 変換を利用する必要がないため, 復号回数はベクトルの次元に依存せず 1 回のみ (約 5 msec) であり, ベクトルの次元が増大するほど, 計算委託比率が向上する.

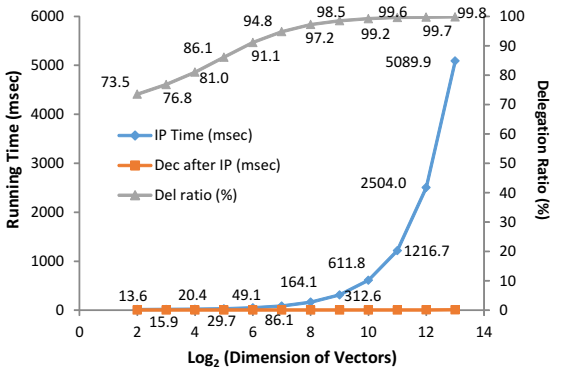

図 4　暗号文ベクトルと平文ベクトルの内積の計算時間

4.3.3　検索済みの暗号文に対する準同型演算

提案 MR-SHE 方式では, 鍵付き準同型 ID ベース暗号方式 [9] の準同型演算鍵を検索用トラップドアとして使用するため, 検索手順と準同型演算アルゴリズム内での暗号文正当性確認手順とは同じ手続きとなる. そのため, 実用上検索済みで同じキーワードに関連付けされていることが保証されている暗号文に対しては, 準同型演算アルゴリズム内で暗号文正当性確認手順を省略することができる. 全体の計算時間のうち, この暗号文正当性確認手順が占める割合を図 5 に示す. mEval, P and C, C and C はそれぞれ MR-SHE.mEval アルゴリズム, 暗号文ベクトルと平文ベクトルの内積, 暗号文ベクトル同士の内積を示す.

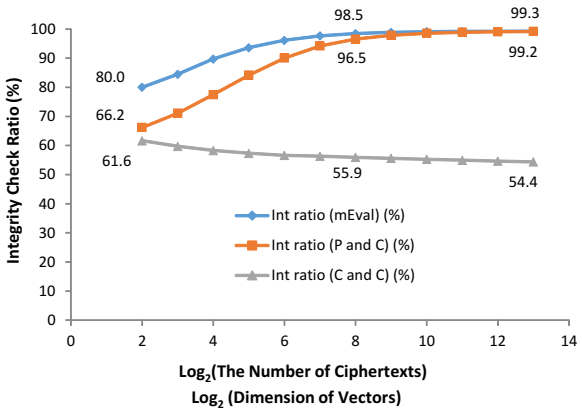

図 5 準同型演算の計算時間に対する暗号文正当性確認の割合

暗号文ベクトル同士の内積では, Catalano-Fiore 変換による準同型乗算の演算コストが準同型加算よりも高いことから, 暗号文正当性確認手順が占める割合が 50%程度であるが, それ以外の場合は準同型加算のみで計算できるため, ほとんどの計算時間を暗号文正当性確認手順が占めていることがわかる. よって, 検索済みの暗号文に対しては, 暗号文正当性確認手順を省略することで, 大幅に効率化することができる.

5. まとめ

本論文では, 誤った準同型演算処理を防止する方策として, 同じキーワードに関連した暗号文に対してのみ準同型演算を許す “まぜるな危険準同型暗号” を提案し, 実装によりその効率性を示した. 提案方式は, Lifted ElGamal の手法と Catalano-Fiore 変換により, 小さな整数のベクトルを平文とする暗号文ベクトル同士の内積演算が可能である.

Catalano-Fiore 変換により, 内積値の復号処理時間はベクトルの次元に比例する. このため, 8000 次元程度であれば 70 秒程度で復号できるものの, より大きな次元のベクトルについては復号が困難になる. より大規模なデータベースを考えるためには, Catalano-Fiore 変換を必要としない, 加法準同型と乗法準同型を同時に満たす “まぜるな危険準同型暗号”が必要である.

謝辞: 本研究は科研費 24700009, 15K00028, および JST CREST の助成を受けたものです.

参考文献

[1] The PBC (Pairing-Based Cryptography) library. Available at `http://crypto.stanford.edu/pbc/`.

[2] Abdalla, M., Bellare, M., Catalano, D., Kiltz, E., Kohno, T., Lange, T., Malone-Lee, J., Neven, G., Paillier, P. and Shi, H.: Searchable Encryption Revisited: Consistency Properties, Relation to Anonymous IBE, and Extensions, *J. Cryptology*, Vol. 21, No. 3, pp. 350–391 (2008).

[3] Abdalla, M., Bellare, M. and Neven, G.: Robust Encryption, *TCC*, pp. 480–497 (2010).

[4] Barak, B., Goldreich, O., Impagliazzo, R., Rudich, S., Sahai, A., Vadhan, S. P. and Yang, K.: On the (im)possibility of obfuscating programs, *J. ACM*, Vol. 59, No. 2, p. 6 (2012).

[5] Boneh, D., Crescenzo, G. D., Ostrovsky, R. and Persiano, G.: Public Key Encryption with Keyword Search, *EUROCRYPT*, pp. 506–522 (2004).

[6] Boneh, D., Segev, G. and Waters, B.: Targeted malleability: homomorphic encryption for restricted computations, *Innovations in Theoretical Computer Science*, pp. 350–366 (2012).

[7] Catalano, D. and Fiore, D.: Using Linearly-Homomorphic Encryption to Evaluate Degree-2 Functions on Encrypted Data, *ACM CCS*, pp. 1518–1529 (2015).

[8] Cramer, R., Gennaro, R. and Schoenmakers, B.: A Secure and Optimally Efficient Multi-Authority Election Scheme, *EUROCRYPT*, pp. 103–118 (1997).

[9] Emura, K., Hanaoka, G., Nuida, K., Ohtake, G., Matsuda, T. and Yamada, S.: Chosen Ciphertext Secure Keyed-Homomorphic Public-Key Encryption, *IACR Cryptology ePrint Archive*, Vol. 2013, p. 390 (2013).

[10] Emura, K., Hanaoka, G., Ohtake, G., Matsuda, T. and Yamada, S.: Chosen Ciphertext Secure Keyed-Homomorphic Public-Key Encryption, *Public-Key Cryptography*, pp. 32–50 (2013).

[11] Gentry, C.: Practical Identity-Based Encryption Without Random Oracles, *EUROCRYPT*, pp. 445–464 (2006).

[12] Gentry, C.: Fully homomorphic encryption using ideal lattices, *STOC*, pp. 169–178 (2009).

[13] Gentry, C., Sahai, A. and Waters, B.: Homomorphic Encryption from Learning with Errors: Conceptually-Simpler, Asymptotically-Faster, Attribute-Based, *CRYPTO*, pp. 75–92 (2013).

[14] Izabachène, M. and Pointcheval, D.: New Anonymity Notions for Identity-Based Encryption, *SCN*, pp. 375–391 (2008).

[15] Jutla, C. S. and Roy, A.: Dual-System Simulation-Soundness with Applications to UC-PAKE and More, *ASIACRYPT*, pp. 630–655 (2015).

[16] Kiltz, E.: Chosen-Ciphertext Security from Tag-Based Encryption, *TCC*, pp. 581–600 (2006).

[17] Lai, J., Deng, R. H., Ma, C., Sakurai, K. and Weng, J.: CCA-Secure Keyed-Fully Homomorphic Encryption, *Public-Key Cryptography*, pp. 70–98 (2016).

[18] Libert, B., Peters, T., Joye, M. and Yung, M.: Non-malleability from Malleability: Simulation-Sound Quasi-Adaptive NIZK Proofs and CCA2-Secure Encryption from Homomorphic Signatures, *EUROCRYPT*, pp. 514–532 (2014).

[19] Lu, W., Yamada, Y. and Sakuma, J.: Efficient Secure Outsourcing of Genome-Wide Association Studies, *IEEE Symposium on Security and Privacy Workshops*, pp. 3–6 (2015).

[20] Shimizu, K., Nuida, K., Arai, H., Mitsunari, S., Attrapadung, N., Hamada, M., Tsuda, K., Hirokawa, T., Sakuma, J., Hanaoka, G. and Asai, K.: Privacy-preserving search for chemical compound databases, *Bioinformatics*, Vol. 16, No. 18 (2015).

[21] Zhang, R. and Imai, H.: Combining Public Key Encryption with Keyword Search and Public Key Encryption, *IEICE Transactions*, Vol. 92-D, No. 5, pp. 888–896 (2009).