高機能暗号を活用した情報漏えい対策「暗号化状態処理技術」の最新動向

清藤武暢（せいとうたけのぶ）／四方順司（しかたじゅんじ）

要　旨

金融分野をはじめとする企業や組織等では、近年、クラウドサービス等の外部サーバを利用するケースが増加している。通常、外部サーバを利用する場合には、同サーバ上に保管されるデータは、安全性確保の観点から暗号化することによって保護している。しかし、暗号化したデータを利用するには一度復号する必要があり、多くの場合、サーバ上で復号されている事情とあわせて考えると、（1）攻撃者による外部サーバへの不正アクセスや（2）外部サーバ管理者等の内部関係者による不正等の強い脅威を想定した場合、暗号化による対策だけでは情報漏えいを防ぎきることは難しいといえる。その意味で、これらの脅威への対策は喫緊の課題といえるが、近年、データを暗号化した状態のまま処理することで、情報漏えいを防ぐというコンセプトの技術「暗号化状態処理技術」の研究が活発化しており、既に製品も登場している。同技術は、共通鍵暗号方式（AES 等）や公開鍵暗号方式（RSA、楕円曲線暗号等）といった従来の基礎的な暗号技術よりも高度な機能を実現可能である反面、実現可能な機能や安全性の基準等が複雑であり、同技術に馴染みのない企業等が利用するには、ハードルが高い状況にあるといえる。そこで、本稿では、暗号化状態処理技術の概要や研究動向を中心に紹介するとともに、同技術に利用する際の留意点等について考察する。

キーワード：暗号化状態処理技術、高機能暗号、秘匿検索、秘匿計算、代理人再暗号化、ペアリング技術／暗号、情報漏えい

本稿の作成に当たっては、独立行政法人産業技術総合研究所研究員の松田隆宏氏から有益なコメントを頂いた。ここに記して感謝したい。ただし、本稿に示されている意見は、筆者たち個人に属し、日本銀行および国立大学法人横浜国立大学の公式見解を示すものではない。また、ありうべき誤りはすべて筆者たち個人に属する。

清藤武暢　日本銀行金融研究所（E-mail: takenobu.seitou@boj.or.jp）
四方順司　国立大学法人横浜国立大学大学院環境情報研究院
（E-mail: shikata@ynu.ac.jp）

1. はじめに

近年、金融機関を含む企業・組織等では、システムの導入や運用に要するコストの削減を主な目的として、クラウドサービス等の外部サーバを業務に利用するケースが増加している。金融機関に限ってみると、約 2 割がクラウドサービスを利用しているとの調査結果がある（2013 年 3 月時点、金融情報システムセンター：FISC［2014］）。こうした外部サーバにおける一般的な情報漏えい対策としては、保管されるデータの暗号化（例えば、FISC［2012］[1] 技 28，29）、データへのアクセス制御（同技 43，44）、規程等の運用ルールの徹底（同運 1-6, 10）等が挙げられる。しかし、インターネットからの不正アクセスによる情報漏えいが多数発生していることに加えて、インシデントの約 1 割が内部関係者の不正に起因するとの調査結果があり[2]、運用ルールが遵守されないケースも窺えるため、こうした対策だけでは情報漏えいを防ぎ切れないのが実情といえる。

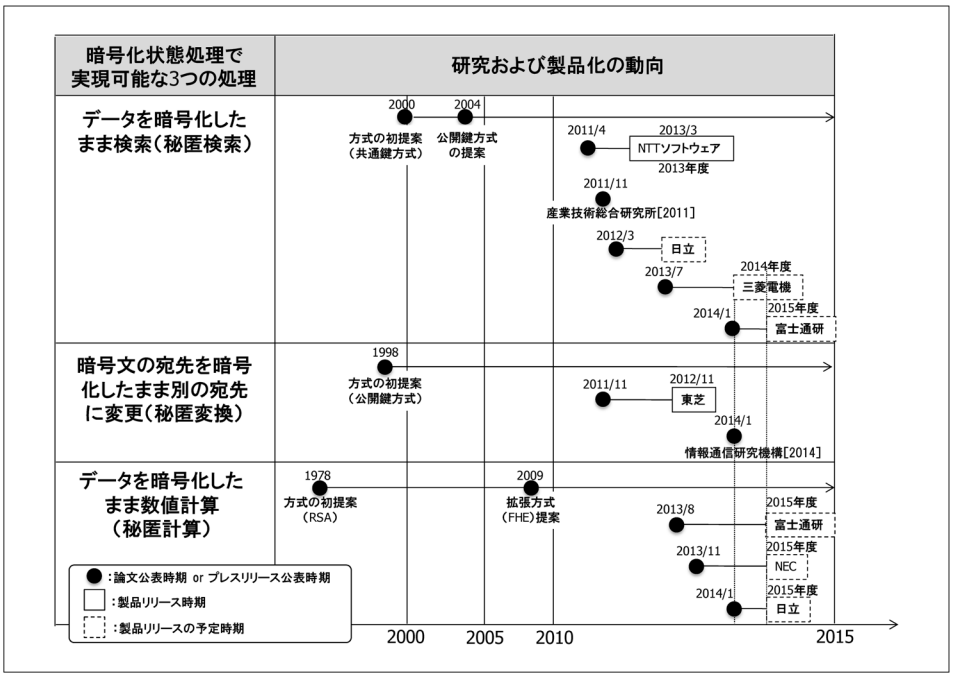
こうした状況を踏まえると、外部サーバに預けるデータの機密性が高い場合には、従来の情報漏えい対策では防ぐことが困難な状況を想定して対策を考えることも有用である。例えば、外部サーバに保管されるデータの暗号化という対策については、通常は暗号化された状態であるものの、利用時には一時的に復号され、平文の（暗号化が解かれた）データがメモリやストレージ等に現れるという特徴がある。このため、攻撃者が不正アクセス等により同サーバの任意のデータにアクセス可能な権限を奪取していれば、同メモリ等にアクセスすることで対象とするデータを盗取できる。近年、メモリ上に現れる復号されたデータの盗取を目的としたマルウエアが発見されるという事例も発生しており、同脅威はより現実的なものとなっている（Verison Business [2009]）。

学界の研究成果をみると、こうした強力な攻撃者を想定しても、サーバからの情報漏えいを防止可能な技術的対策が 2000 年以前から提案されている（図表 1）。すなわち、「データを暗号化した状態でサーバに預け、サーバは、利用者の指示に基づき、データを暗号化した状態のまま処理を行い、その結果を利用者に返すという技術」である。同技術については、さまざまな研究が存在するほか、既に一部では電子メールシステムやファイル共有システム等に応用した製品も登場しているものの、同技術を総称する呼称や定義について学界ではまだコンセンサスが形成されていない状況にある。そこで、本稿では、「データを暗号化したままの状態で処理する技術」を「暗号化状態処理技術」と呼ぶことにする。暗号化状態処理技術は、

1 わが国の金融機関は、情報セキュリティ対策を講じる際、「金融機関等コンピュータセキュリティシステムの安全対策基準・解説書」（FISC［2012］）を指針としている。

2 米大手情報セキュリティ企業 SafeNet 社が、全世界で発生した情報漏えい等に関する統計情報を公表している（「Breach Level Index」、http://www.breachlevelindex.com）。

図表 1　暗号化状態処理技術の初期提案の時期と製品化動向



共通鍵暗号方式[3]や公開鍵暗号方式[4]といった従来からある基礎的な暗号技術よりも高度な機能を実現可能である反面、実現可能な機能や安全性の基準等が複雑であり、同技術に馴染みのない企業等が利用するには、ハードルが高い状況にあるといえる。そこで、本稿では、同技術について、実現可能な処理内容、具体的な応用例、安全性に関する研究動向等を紹介するとともに、利用上の留意点に関する検討も行う。

以下、本稿では、2 節において、クラウド等の外部サーバを利用する際のモデルを示したうえで、データを単に暗号化するという従来の対策の限界を説明する。3 節において、暗号化状態処理技術の概要を紹介したうえで、4 節で、同技術における代表的な 3 つの技術について説明する。5 節では、暗号化状態処理技術の利用上の留意点について考察する。

3 共通鍵暗号方式は、暗号化に利用する鍵（後述の暗号化鍵）と復号に利用する鍵（同復号鍵）が同一（共通）であり、この鍵を当事者が秘密に保管する必要がある。また、暗号化／復号処理が後述の公開鍵暗号方式よりも相対的に高速である。例えば、米国政府標準暗号である「AES（Advanced Encryption Standard）」等が知られている。

4 公開鍵暗号は、暗号化鍵（公開鍵）と復号鍵（秘密鍵）が異なるほか、暗号化鍵から復号鍵を求めることが計算量的に困難であるため、暗号化鍵を公開できる。例えば、RSA や楕円曲線暗号等がある。

2. 外部サーバ利用時における脅威の整理

本節では、まず、企業等が外部サーバを利用して業務を行う際の典型的なモデルを示したうえで、同モデルにおいて想定すべき脅威や安全性要件を整理する。また、従来の単なるデータの暗号化では情報漏えいを防ぎ切れないことを説明する。

(1) 検討対象とする想定モデル

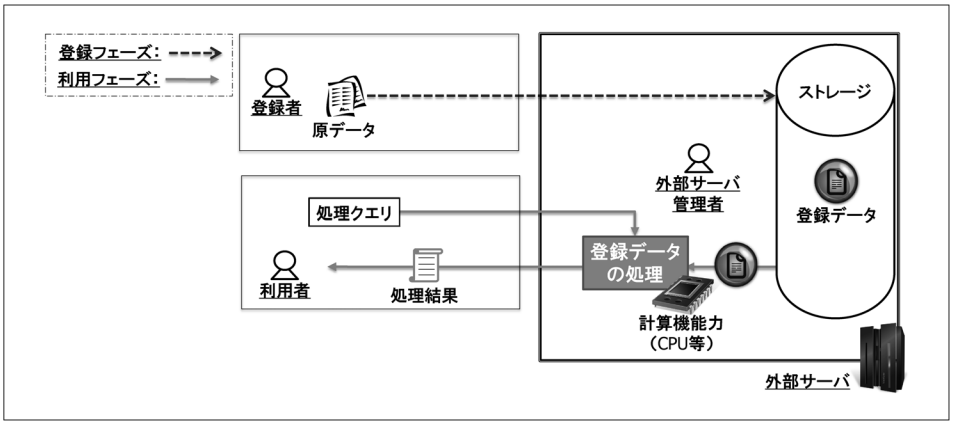
企業等が外部サーバを利用する形態は多岐にわたるが、本稿では、「登録者」、「利用者」、「外部サーバ」、「外部サーバ管理者」、の4エンティティから構成される単純なモデルを想定する。各エンティティの概要は以下のとおりである（図表2）。

登録者：自前の端末を用いて、業務で使用するデータ（以下、「原データ」と呼ぶ）を外部サーバに登録するエンティティ。以下では、登録されたデータのことを「登録データ」と呼ぶ。

利用者：自前の端末を用いて、登録データに対する処理要求（以下、「処理クエリ」と呼ぶ）を外部サーバに送信し、その「処理結果」を受け取るエンティティ。個別の応用例によっては、登録者と利用者が同一エンティティの場合や利用者が複数存在する場合がある。

外部サーバ：ストレージや計算機能力（CPUやメモリ等）を有するサーバであり、登録データを保管するほか、利用者からの処理クエリを受けて、データ処理を行い、その「処理結果」を利用者に返信する。なお、外部サーバは、登録者

図表2　想定モデルにおける処理フロー



や利用者の端末と適切に暗号通信[5]を行っており、通信路上でデータを盗聴・改ざんされないとする。

外部サーバ管理者：外部サーバの管理を行うエンティティであり、外部サーバのストレージに保管されたすべてのデータだけでなく、同サーバ内でのデータ処理中に生成されるすべての中間データにもアクセス可能な権限（以下、「管理者権限」と呼ぶ）を有する。

(2) 想定する脅威と求められる安全性

次に、想定モデルにおける脅威と求められる安全性について整理する。企業等が利用する外部サーバにおいては、攻撃者の不正アクセスによる管理者権限の奪取や、正規の外部サーバ管理者による不正等が想定される。いずれの場合においても、外部サーバに保管されたすべてのデータや処理中にメモリ等に現れる中間データ等を盗取される可能性がある。そこで、本稿では、攻撃者は外部サーバの管理者権限を有すると仮定する（攻撃者 1）。また、個別の応用例によっては利用者が複数存在するため、その場合には攻撃者 1 が一部の利用者と結託することも考えられる（攻撃者 2）。攻撃者 1 よりも攻撃者 2 の方が強力な攻撃者であり、攻撃者 2 に対して後述の安全性要件を満たす場合には、攻撃者 1 に対しても同安全性要件を満たすことは明らかである。なお、攻撃者がすべての利用者と結託する場合には、そもそも業務が遂行できなくなるため、そうした状況は想定しないほか、外部サーバからの情報漏えいを検討対象とするため、外部サーバへのデータ登録を行う登録者は不正を行わないとする。

攻撃者 1：外部サーバの管理者権限を有しており、同サーバに保管されたすべてのデータやデータ処理中の中間データを盗取可能。結託は行わない。
攻撃者 2：外部サーバの管理者権限を有するほか、一部の利用者と結託する。

また、求められる安全性については、企業等が外部サーバに預けたデータの保護の観点から、想定する攻撃者に対して原データを漏えいしないことが挙げられる。なお、本稿では、情報漏えい対策を検討対象とするため、登録データや処理結果の改ざん等は検討対象外とする[6]。

5 暗号通信を行うためには、VPN（Virtual Private Network）や SSL（Secure Socket Layer）／TLS（Transport Layer Security）等の暗号技術を利用可能である。

6 外部サーバ上の原データの一貫性や処理の正当性等を登録者や利用者が確認する方法としては、例えば、電子署名や耐タンパ性を有するハードウエアを利用して、原データの処理に関するログ等のデータを生成・保管しておく方法がいくつか提案されている。暗号化状態処理技術においても、一貫

安全性要件：想定する攻撃者に対して原データを漏えいしない。

(3) 従来のデータ暗号化による対策の限界

外部サーバのストレージに保管されたデータを保護するためには、基本的な暗号技術（AES、RSA 等）を用いてデータを暗号化するという対策が、従来から採用されてきた。そこで、同対策を実施することで、想定する攻撃者に対して原データを守秘できるか否か（安全性要件）について評価する。同対策では、原データの登録時には、外部サーバは、登録者から受信した原データを暗号化用の鍵（以下、「暗号化鍵」と呼ぶ）で暗号化したうえで登録データとしてストレージに保管する。また、原データの利用時には、同サーバは、ストレージから読み出した登録データを復号用の鍵（以下、「復号鍵」と呼ぶ）で一時的に復号し、利用者が指定したデータ処理を行う。なお、暗号化／復号鍵は、同サーバに保管されている。

こうした対策に対して攻撃者 1（管理者権限のみ）を想定すると、同攻撃者は、①一時的に復号され、メモリ等に現れた原データにアクセスする方法、あるいは、②復号鍵を不正に使用して登録データを復号する方法等により原データを盗取可能である。よって、同対策は、攻撃者 1 に対して安全性要件を満たさないといえる。これを受けて、学界では、攻撃者 1、さらには、攻撃者 2（管理者権限 + 結託あり）に対しても安全性要件を満たす技術（暗号化状態処理技術）が活発に研究されている。3 節では、同技術の概要を紹介する。

3. 暗号化状態処理技術

本節では、暗号化状態処理技術の概要や脅威等について解説する。

(1) 暗号化状態処理技術の概要

暗号化状態処理技術については、その呼称や定義に関する学界のコンセンサスが形成されているとは言い難いが、本稿では「データを暗号化したままの状態で処理

性と処理の正当性の保証が課題として指摘されている（Wang *et al.* [2009]、van Dijk and Juels [2010] 等）。

する技術の総称」と定義する。同技術では、原データを暗号化した状態で外部サーバに預け、同サーバが暗号化したままデータ処理を行うため、同サーバから原データが漏えいしないというメリットがある。学界では、暗号化したままデータの「乗算と加算」を実行可能な方式が提案されており（Gentry [2009]）、処理効率を気にしなければ理論的には乗算と加算に基づく任意のデータ処理を暗号化したまま実行可能である[7]。このような状況のなか、多くの研究グループは、処理内容を限定する代わりに現実的な時間で実行可能な暗号化状態処理を実現する方式を盛んに研究している。これらの研究において実現されている暗号化状態処理として、主に次の3つが挙げられる。

秘匿検索[8]：原データを暗号化したままキーワード検索を行う処理、または、同処理を実現する技術。同技術は、大量のデータ（電子メール等）を暗号化したうえで外部サーバに登録しておき、利用者が指定したキーワードについて、同サーバに検索を行わせる用途等が想定されている。

秘匿変換[9]：登録者のみが復号可能な暗号文（以下、「○○宛の暗号文」と表記）を、一度も復号することなく、利用者宛の暗号文に変換する処理、または、同処理を実現する技術。同技術は、将来、利用者と共有する可能性のあるファイル（原データ）を、登録者宛の暗号文として暗号化しておき、その後、同ファイルを共有したい任意の利用者宛の暗号文（処理結果）に変換することで、同ファイルを共有する用途等が想定されている。

秘匿計算[10]：データを暗号化したまま数値計算等を行う処理、または、同処理を実現する技術。同技術は、大量のデータ（購入履歴、患者情報等）を暗号化したうえで外部サーバに登録しておき、利用者が指定した統計解析等の数値計算を同サーバに行わせる用途等が想定されている。

(2) 暗号化状態処理技術の典型的な処理フロー

上記の各暗号化状態処理技術に共通する典型的な処理フローを示す（図表3）。同技術は、登録データの登録・利用に用いる3種類の鍵（以下、それぞれ、「登録

7「乗算と加算」が演算可能であれば、乗算と加算によりそれぞれ除算と減算も演算可能であり、実質的に「四則演算」が可能である。暗号化したまま四則演算が可能な方式を利用すれば、例えば、「暗号化鍵を用いたAESの暗号化処理」を、外部サーバに暗号化鍵を漏えいすることなく委託できることが示されている（Gentry, Halevi, and Smart [2012]）。

8 学界では、「秘匿変換」の実現方式は「秘匿検索暗号」や「検索可能暗号」等と呼ばれる。

9 学界では、「秘匿変換」および「秘匿変換における処理鍵」は、「代理人再暗号化」および「再暗号化鍵」とそれぞれ呼ばれる。

10 学界では、「秘匿計算」の実現方式の1つとして「準同型暗号」が研究されている。

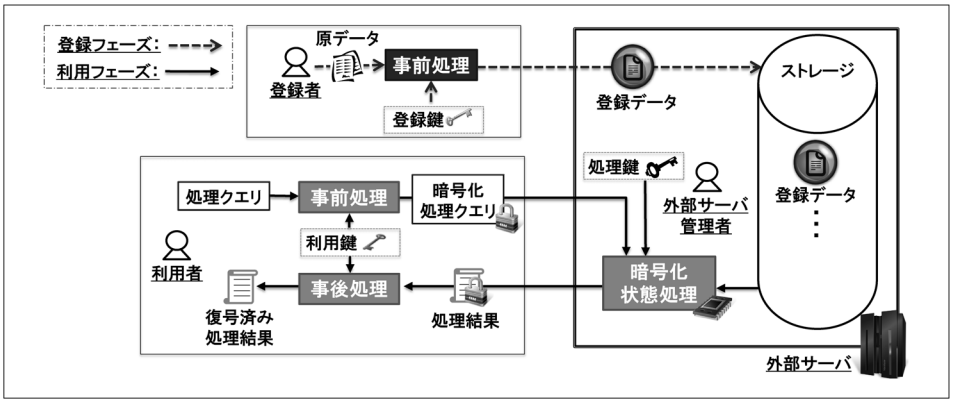
鍵」、「利用鍵」、「処理鍵」と呼ぶ）を生成する「鍵生成フェーズ」、登録データを登録する「登録フェーズ」、登録データを利用する「利用フェーズ」の 3 つから構成される。なお、より詳細な処理フローは各技術に依存する。

鍵生成フェーズ：いずれの技術においても、利用者が利用鍵を生成する。秘匿検索と秘匿計算については、利用者が利用鍵に対応する登録鍵を生成する。また、秘匿変換については、登録者が登録鍵を生成し、登録者と利用者が協力して「登録鍵と利用鍵」に対応する処理鍵を生成する。登録鍵、利用鍵、処理鍵は、登録者、利用者、外部サーバ管理者がそれぞれ保管する（図表 4）。

登録フェーズ：登録者は、登録鍵を用いて原データに「事前処理」を施したうえで、外部サーバに送る。外部サーバは、登録者から受信したデータを登録データとしてそのままストレージに保管する（図表 3）。

利用フェーズ：利用者は、まず、処理クエリを生成し、これに利用鍵を用いて事前処理を施すことで「暗号化処理クエリ」を生成したうえで、外部サーバに送信する。外部サーバは、暗号化処理クエリに基づき処理鍵を用いてデータ処理を行い、その処理結果を利用者に返す。利用者は、受信した処理結果に利

図表 3　登録データの登録・利用に関する処理フロー



図表 4　各鍵を生成または所持するエンティティ

		登録鍵	利用鍵	処理鍵
鍵の生成者	秘匿検索	利用者	利用者	なし
	秘匿変換	登録者		登録者と利用者
	秘匿計算	利用者		なし
鍵の所有者		登録者	利用者	外部サーバ管理者

用鍵を用いて事後処理を施すことで「復号済み処理結果」を生成し、結果を得る（図表 3）。なお、処理鍵の利用の有無や、事前／事後処理の有無は各技術に依存するため、詳細は 4 節を参照のこと。

(3) 暗号化状態処理技術において想定される脅威

暗号化状態処理技術において想定する攻撃者と安全性要件を確認する。同技術では、外部サーバに処理鍵が保管されており、攻撃者 1（管理者権限のみ）は、処理鍵も含めて同サーバ内のすべてのデータにアクセス可能である。また、いずれの暗号化状態処理技術でも利用者が利用鍵を有しており、攻撃者 2（管理者権限 + 結託あり）は、利用者と結託することで当該利用者の利用鍵も使用可能になる。ただし、処理鍵を用いない技術（秘匿検索、秘匿計算）では、攻撃者による同鍵の利用を想定しないほか、利用者が 1 人しか想定されない場合には、攻撃者と利用者の結託（攻撃者 2）を想定しない。

4. 暗号化状態処理の代表的な技術

本節では、現在実現されている主な 3 つの暗号化状態処理である秘匿検索、秘匿変換、秘匿計算を取り上げ、実現アイデアや処理フロー、代表的な応用例、安全性等に関する研究動向の観点からそれぞれ説明する。

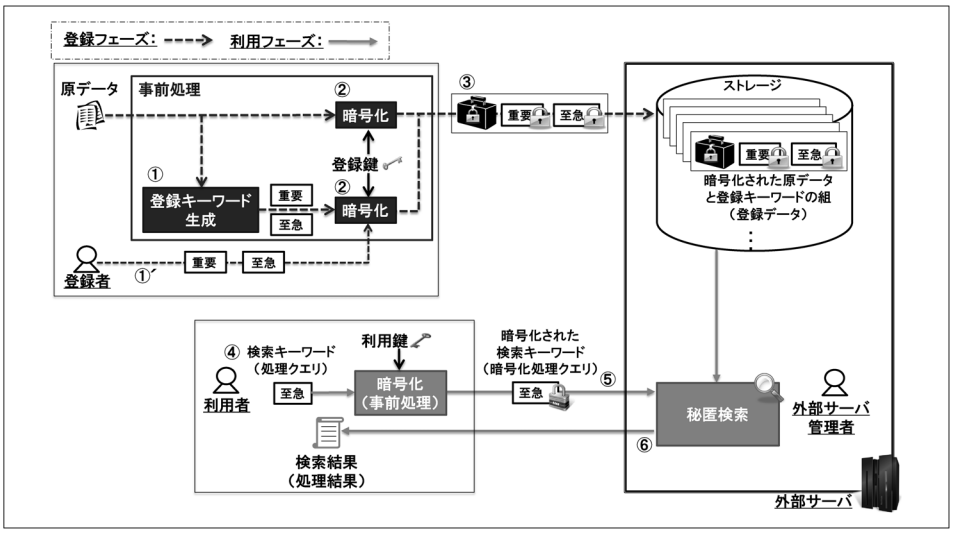
(1) 秘匿検索

イ. 実現アイデアと処理フロー

秘匿検索は、前述のとおり、原データを暗号化したままキーワード検索を行う技術である。同処理を実現するためには、予め、原データとは別にキーワード（以下、「登録キーワード」と呼ぶ）を設定し、同キーワードを暗号化して登録しておく必要がある。秘匿検索を実現する方法は多数存在するが、例えば、検索用のキーワード（以下、「検索キーワード」と呼ぶ）を暗号化した際に、登録キーワードと検索キーワードが同一であれば、それらの暗号文も同一になるような暗号化方式[11]

11 このような特徴を有する暗号化方式は、特に「確定的暗号」と呼ばれる。これに対して、同じ平文に対して毎回異なる暗号文が生成されるような暗号方式は「確率的暗号」と呼ばれる。

図表 5　秘匿検索の処理フロー



（例：AES 等の共通鍵暗号方式）を用いることで、キーワードを平文の状態で比較する代わりに暗号文同士を比較しても検索が可能になる。なお、同技術における現在主流の実現方式では、原データの中に検索キーワードが含まれていたとしても、予め登録キーワードとして設定していない場合には検索されないという制約がある[12]。以下では、各鍵を生成する処理（鍵生成フェーズ）、原データを外部サーバへ登録する処理（登録フェーズ）、キーワード検索を行う処理（利用フェーズ）の処理フローを示す（図表 5）。

鍵生成フェーズ：利用者は、暗号化鍵／復号鍵を生成し、復号鍵を利用鍵として保管する。暗号化鍵を登録鍵として登録者に預託する。

登録フェーズ：登録者は、まず、原データに関する登録キーワード（例えば、「重要」や「至急」等）を設定する。登録キーワードの設定方法として、「原データから自動的に登録キーワードを抽出する方法[13]（図表 5-①）」と「登録者が手動で設定する方法（同①′）」が挙げられる。そして、登録者は登録鍵を使用して、登録キーワードについては秘匿検索技術により暗号化し、原データについては AES 等の通常の暗号技術により暗号化する（同②）。この①②が、登

12 登録キーワードを設定しなくとも、原データの暗号文に対してキーワード検索が可能な方式が提案されている（小暮ほか［2014］）。

13 例えば、原データにおいて出現頻度の高い単語を登録キーワードとして設定する方法や、「重要」や「至急」等の単語を登録キーワードの候補としてデータベースに登録しておき、当該単語が原データに 1 度でも出現した場合に、登録キーワードとして設定する方法等が考えられる。

録者の事前処理に相当する。そして、登録者は、暗号化された登録キーワードと暗号化された原データを合わせて登録データとして、外部サーバに保管する（同③）。なお、前述の実現アイデアで示した例では、原データだけでなく登録キーワードについても、通常の暗号技術（AES 等）を秘匿検索技術として用いて暗号化を行っている[14]。また、5 節（1）で触れるが、原データの暗号化については、通常の暗号技術だけでなく、秘匿変換技術や秘匿計算技術による暗号化を行うことも可能である。

利用フェーズ：利用者は、まず、検索キーワード（処理クエリ）を生成する（同④）。次に、利用鍵を用いて同キーワードを秘匿検索技術により暗号化（事前処理）することにより暗号化処理クエリを生成したうえで、これを外部サーバに送信する（同⑤）。外部サーバは、ストレージから暗号化された登録キーワードを順次読み出し、受信した暗号化処理クエリとの比較を行い、一致するものがあったか否か等の情報（処理結果）を利用者に返す（同⑥）。なお、利用者は事後処理を行わないほか、外部サーバが該当する登録データを利用者に送信するか否かは個別のシステム要件に依存する。

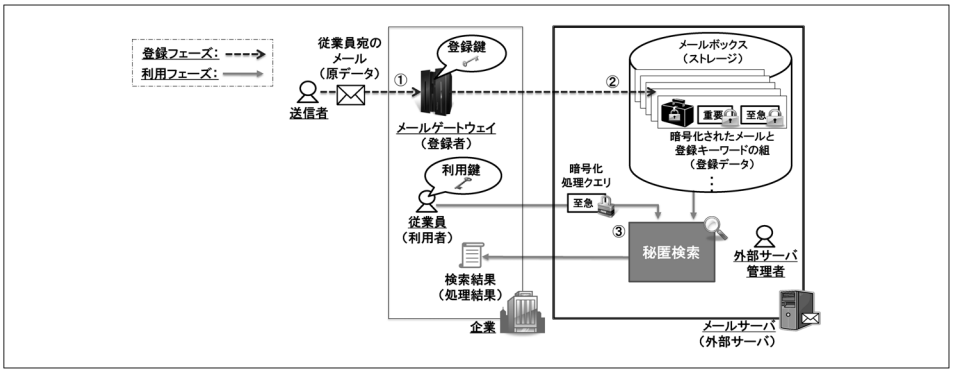
ロ．代表的な応用例

秘匿検索の代表的な 2 つの応用例を紹介する。1 つは、電子メールシステムに関する事例であり、外部サーバに預けた電子メールを暗号化したまま秘匿検索するというものである（Boneh *et al.* [2004]、NTT ソフトウェア［2013］等）。具体的には、企業の従業員宛に外部の送信者から電子メールが送信されてきた際に（図表 6-①）、同企業に設置されたメールゲートウェイ（登録者）が、同電子メールに対して自動で登録キーワード（例：重要、至急）を設定し、これらを暗号化したうえで外部のメールサーバのメールボックス（外部サーバ）に保管する（同②）。従業員は、自分のメールボックスから暗号化された電子メールを読み出し（復号したうえで）閲覧することが可能であるほか、利用鍵を用いて検索キーワードに関する秘匿検索を要求し、検索結果（処理結果）を得ることもできる（同③）。なお、登録鍵を公開することで、メールゲートウェイの代わりに送信者がメールボックスへの登録作業を行うことも考えられる。

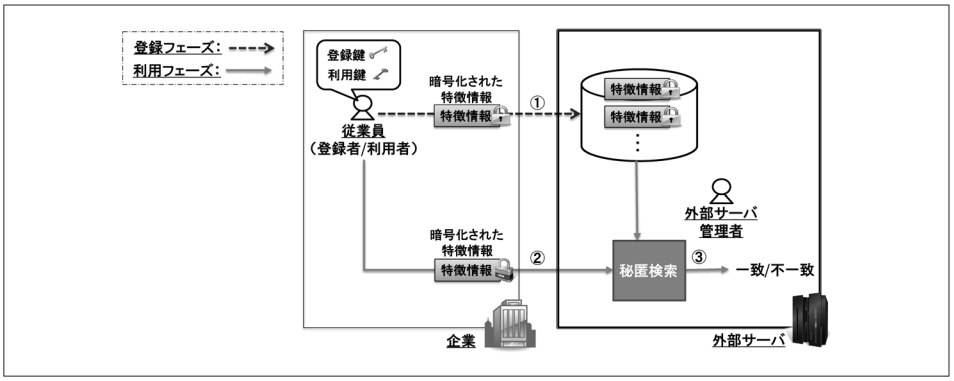
もう 1 つは、ATM 取引における顧客の本人確認等に利用されている生体認証システムへの応用例である（富士通研究所［2013］）。生体認証に用いる生体情報は、機微な個人情報であることから、金融業界では各顧客の IC キャッシュカードに保管して保護する形態が一般的となっているものの、従業員向けシステム等ではサーバで一元管理を行っているケースも存在する。この場合、同サーバへの不正アクセ

14 この例は、非常にシンプルな実現方式であり、後述の安全性要件 a3 を満たさないほか、部分一致検索や類似検索を実現することはできない。

図表 6　秘匿検索の電子メールシステムへの応用例



図表 7　秘匿検索の生体認証システムへの応用例



ス等により全従業員の生体情報が漏えいするリスクがある。同応用例では、サーバに登録した生体情報を暗号化したまま本人確認に利用することで、同サーバから生体情報が漏えいすることを防止するというものである（「テンプレート保護技術」と呼ばれる）。具体的には、企業の従業員（登録者／利用者）は、自分の生体情報（原データ）から特徴情報（登録キーワード）を生成し、これを暗号化したデータ（登録データ）をサーバ（外部サーバ）に登録する（図表 7-①）。認証時には、改めて取得した自分の生体情報から特徴情報（検索キーワード）を生成し、これを暗号化したうえでサーバに送信する（同②）。サーバは、受信したデータと登録データの照合（秘匿検索）を行い、一致／不一致を照合結果（処理結果）として返す（同③）。

ハ. 研究動向

（イ） 学界で検討対象とされる安全性要件と攻撃者

学界では、原データの保護（安全性要件）のほかに、秘匿検索に固有の 3 つの安全性要件（安全性要件 a1～a3）も検討対象として取り上げている。具体的には、外部サーバには、原データの暗号文だけでなく、登録キーワードの暗号文も預けていることから、想定する攻撃者に対して登録キーワードを漏えいしないことを要件として取り上げている（安全性要件 a1）。また、秘匿検索では、通常、一致するものがあったか否かの情報（検索結果）だけは外部サーバに漏えいする[15]。このため、攻撃者が検索キーワードを入手できれば、登録データの中に同検索キーワードに一致するものがあるか否かを知ることができる。このようにして原データに関する情報が漏えいすることを防止するため、想定する攻撃者に対して検索キーワードを漏えいしないことを要件として取り上げている（安全性要件 a2）。同要件は、直感的には、暗号化処理クエリから検索キーワード（処理クエリ）を求められないという要件である。

しかし、安全性要件 a2 を満たしていても、ある 2 つの暗号化処理クエリが同一の検索キーワードに対応しているか否かを識別可能な場合、攻撃者は、何らかの検索キーワードが頻繁に検索されている等の情報を入手することができる。こうした頻度情報を用いれば、例えば、秘匿検索を行っていない他の検索サービスにおいて頻繁に検索されているキーワード等を参考に、検索キーワードを推測するといった攻撃も想定される[16]。このため、想定する攻撃者に対して、2 つの暗号化処理クエリにそれぞれ対応する検索キーワードが同一か否かの情報を漏えいしないことも要件として取り上げている（安全性要件 a3）。学界では、原データの保護（安全性要件）に加えて安全性要件 a1～a3 も検討対象とした研究が主流となっている。

このほか、想定する攻撃者に関しては、理論的には攻撃者が利用者と結託しない場合（攻撃者 1）と結託する場合（攻撃者 2）がありうる。しかし、既存研究をみると、利用者が 1 人のみであり結託は行わないという攻撃者（攻撃者 1）を検討対象とした研究が主流となっている[17]。

15 なお、一致しているか否かの判定を利用者側で行うことで、検索結果も外部サーバ管理者から保護するという方式も提案されている（小暮ほか［2014］）。同方式では、登録データが多いほど、外部サーバから利用者への通信量や利用者の端末における判定処理負荷が増加するという特徴がある。

16 頻度情報を手掛かりに攻撃対象の情報を推測する攻撃は「頻度分析」と呼ばれる。

17 複数の利用者の一部と結託する攻撃者（攻撃者 2）を想定した安全性評価を行っている研究も存在する（Hwang and Lee [2007]、Hattori *et al*. [2013]）。

安全性要件 a1：想定する攻撃者に対して、登録キーワードを漏えいしない。
安全性要件 a2：想定する攻撃者に対して、検索キーワードを漏えいしない。
安全性要件 a3：想定する攻撃者に対して、2 つの処理クエリにそれぞれ対応する検索キーワードが同一か否かの情報を漏えいしない。

（ロ） 利用する暗号技術による分類

秘匿検索は、同技術の実現に利用する暗号技術の観点から「共通鍵暗号方式ベース」と「公開鍵暗号方式ベース」に大別される（図表 8）。共通鍵暗号方式ベースの秘匿検索（Song, Wagner, and Perrig [2000] 等）では、登録鍵と利用鍵が同一であるため、原データの登録が可能な登録者は、利用者として原データの検索も可能になる（逆も同様）。このため、不特定多数の利用者に利用させるという用途には適さないと考えられている。ただし、公開鍵暗号方式ベースの方式よりも処理速度が相対的に速いため、大規模データベース等のように登録データの数が多いケースに適すると考えられている。公開鍵暗号方式ベースの秘匿検索（Boneh *et al.* [2004] 等）は、登録鍵を公開できるため、不特定多数のユーザに原データの登録を行わせることができる。例えば、前述した電子メールシステムへの応用例のように、不特定多数の登録者（送信者等）が想定されるケースでの利用に適すると考えられている。

（ハ） 実現可能な検索機能

秘匿検索が提案された当初は、2 つのキーワードが完全に一致しているか否かを検索する「完全一致検索」を実現する方式のみが提案されていた（Song, Wagner, and Perrig [2000]、Boneh *et al.* [2004] 等）。その後、さまざまな検索機能を実現する方式の研究が進められている（図表 9）。

図表 8　利用する暗号技術による分類とその特徴

実現方式	公開の可否		相対的な処理速度
	登録鍵	利用鍵	
共通鍵暗号方式ベース	不可	不可	高速
公開鍵暗号方式ベース	可	不可	低速

図表 9　秘匿検索において実現されている検索機能

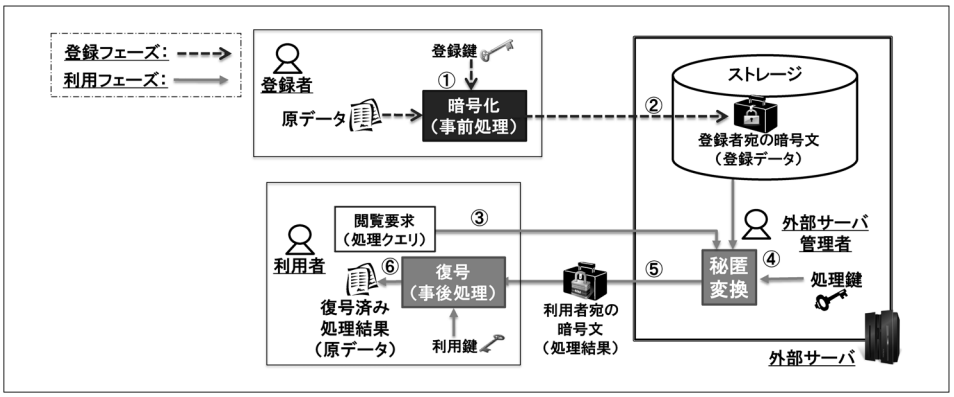
検索機能	概要	主な文献	
		共通鍵暗号方式ベース	公開鍵暗号方式ベース
完全一致検索	検索キーワードと完全に一致するキーワードを有する登録データを検索。	Song, Wagner, and Perrig [2000], Goh [2003], Curtmola *et al.* [2006], Kamara and Papamanthou [2013]	Boneh *et al.* [2004], Abdalla *et al.* [2005], Bellare, Boldyreva, and O'Neill [2007]
部分一致検索	検索キーワードが文字列の一部に含まれるキーワードを有する登録データを検索。なお、同検索には、前方（後方）一致検索も含まれる。	※本項目は、該当する研究成果が現在公表されていない。ただし、今後公表される可能性はある。	吉田・小田・小林［2010］
OR 検索	複数個の検索キーワードの少なくとも 1 つと完全に一致するキーワードを有する登録データを検索。		Baek, Safavi-Naini, and Susilo [2008], Katz, Sahai, and Waters [2013]
AND 検索	複数個の検索キーワード全てと完全に一致するキーワードを有する登録データを検索。	Golle, Staddon and Waters [2004]	Boneh and Waters [2007], Baek, Safavi-Naini, and Susilo [2008], Katz, Sahai, and Waters [2013]
AND 検索と OR 検索の組合せ	AND 検索と OR 検索を組み合わせたより高度な検索条件に一致するキーワードを有する登録データを検索。	Shen, Shi, and Waters [2009], Yoshino *et al.* [2012], Moataz and Shikfa [2013]	Katz, Sahai, and Waters [2013]
類似検索	検索キーワードの文字列と類似したキーワードを有する登録データを検索。	Li *et al.* [2010]	Dong *et al.* [2013]

(2)　秘匿変換

イ．実現アイデアと処理フロー

秘匿変換は、前述のとおり、登録者宛の暗号文を一度も復号することなく、利用者宛の暗号文に変換する技術である。端的に言えば、「登録者宛の暗号文（登録データ）を登録者の復号鍵で復号する処理」と「復号結果（原データ）を利用者の暗号化鍵で暗号化する処理」を一体化することで、外部サーバがこれらの処理（秘匿変換）を実行しても復号結果を入手できないようにする技術である。この処理に用いる鍵（処理鍵）は、当該登録者と当該利用者用に固有の鍵であり、登録者宛の

図表 10　秘匿変換の処理フロー



暗号文（登録データ）を別の利用者宛の暗号文に変換するためには、別途対応する処理鍵を生成する必要がある。

本節では、秘匿変換の実現方式として主流となっている公開鍵暗号方式ベースを想定する[18]。また、後述するように、学界では登録者が複数のケースや登録者が利用者としても振る舞うケースも研究されているが、以下に示す処理フローでは、登録者および利用者がそれぞれ 1 人のみのモデルを想定する（図表 10）。具体的には、処理鍵等を生成する処理（鍵生成フェーズ）、原データの外部サーバへの登録処理（登録フェーズ）、暗号文の宛先変更を行う処理（利用フェーズ）についてそれぞれ処理フローを示す。

鍵生成フェーズ：登録者および利用者は、それぞれ自身の暗号化鍵／復号鍵（それぞれ公開鍵暗号方式の公開鍵／秘密鍵に対応）を生成する。そして、登録者と利用者が協力して「登録者の復号鍵と利用者の暗号化鍵」から処理鍵を生成し、外部サーバに処理鍵を預託する。外部サーバは、この処理鍵を利用することで、暗号文を登録者宛から当該利用者宛に変換できるようになる。また、登録者は自身の暗号化鍵／復号鍵を、利用者も自身の暗号化鍵／復号鍵をそれぞれ登録鍵と利用鍵として保管する。

登録フェーズ：登録者は、登録鍵を用いて原データを暗号化することで自分宛の暗号文（登録データ）を生成し（図表 10-①。事前処理に相当）、これを外部サーバに保管する（同②）。

利用フェーズ：利用者は、閲覧したい登録データに関する閲覧要求（処理クエリ）を外部サーバに送信する（同③）。なお、秘匿変換では、利用者は事前処理を行わないため、暗号化処理クエリも生成されない。外部サーバは、この閲覧

18　共通鍵暗号方式を用いても秘匿変換を実現可能である（具体例は**補論 2** を参照）。

要求で指定された登録データに対して、処理鍵を用いて秘匿変換を行うことで利用者宛の暗号文（処理結果）を生成し（同④）、これを利用者に送信する（同⑤）。利用者は、この暗号文を利用鍵で復号することで（事後処理）、原データ（復号済み処理結果）を入手する（同⑥）。

ロ. 代表的な応用例

秘匿変換の代表的な応用例として、グループ内でファイル（原データ）を共有するシステムに関する事例を紹介する（東芝ソリューション［2011］[19]）。ファイル共有は、複数の人と共同で作業を進めるうえで不可欠な機能であり、既にさまざまな方法が利用されているものの、いずれの方法にも潜在的な情報漏えいリスクあるいは利便性上の課題が存在する（図表 11）。秘匿変換を利用することでこうした問題を解決することが可能である。

具体的には、まず、グループのマネージャー（登録者）がグループ内で共有したいファイル（原データ）を暗号化して（登録データを生成し）、外部サーバに保管する（図表 12-①）。グループメンバー（利用者）を追加する場合には、同メンバーに対応する処理鍵を生成し（同②）、外部サーバに預けておくことで（同③）、すべての登録データを同メンバー宛の暗号文に変換できるようになる[20]。また、グルー

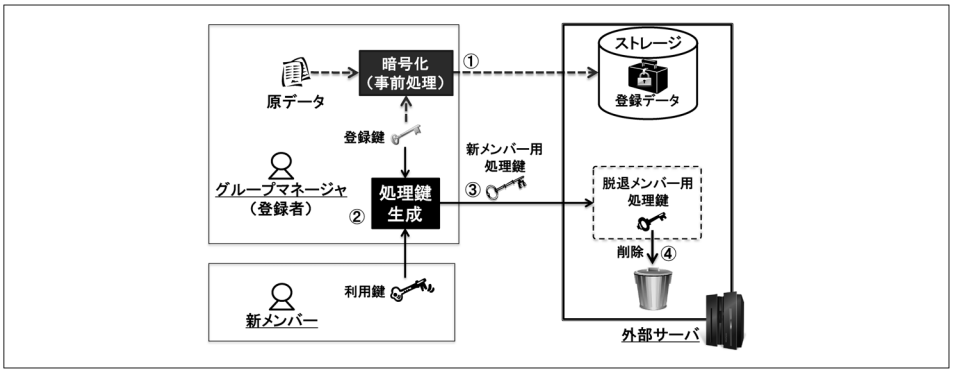
図表 11　既存のファイル共有方法における潜在的リスクや利便性上の課題

既存のファイル共有方法	潜在的リスクや利便性上の課題	秘匿変換の応用による左記の問題の解決
外部サーバにファイルを預け、同サーバが同ファイルへのアクセス制御を行う方法	外部サーバ管理者への情報漏えいのリスクがある。	外部サーバ管理者は、アクセス制御の代わりに、暗号文の変換処理を行うが、その際、原データは漏えいしない。
ファイルの暗号化にグループメンバー間で共通のパスワードや暗号化鍵を用いる方法	メンバーが脱退する度に新しいパスワード等を生成し、同パスワード等で全ファイルを暗号化し直す必要がある。	脱退するメンバーに対応する処理鍵を外部サーバから削除することで、暗号文を同メンバー宛に変換ができなくなる（登録データへの処理は不要）。
グループメンバーごとに異なるパスワードや暗号化鍵等を用いる方法	グループに加入するメンバーが、加入前に生成されたファイルにアクセスするケースを想定すると、メンバー加入ごとに全ファイルを同ユーザのパスワード等で暗号化する必要がある。	加入するメンバーの秘密鍵および対応する処理鍵を生成する。同処理鍵があれば、任意の登録データを同メンバー宛の暗号文に変換できる（登録データへの処理は不要）。

19 同サービスは、2012 年 11 月に開始されたが、2014 年 7 月に終了。

20 各利用者からのファイル閲覧要求に対して、アクセス制御ポリシーに基づき、外部サーバが秘匿変換を行うか否かを判断することで、よりきめ細かいファイル共有が可能になる。

図表 12　秘匿変換のファイル共有システムへの応用例



プメンバーが脱退する場合には、同メンバーに対応する処理鍵を外部サーバから削除することで（同④）、同メンバー宛の暗号文への変換ができなくなる[21]。なお、グループマネージャーの暗号化鍵（公開鍵暗号の公開鍵に対応。登録鍵の一部）をグループ内で共有しておくことで、グループメンバーも登録データ（ただし、登録者宛の暗号文）を登録できるようになる。

ハ．研究動向

秘匿変換に関する研究動向として、変換機能による実現方式の分類と、学界で検討対象とされる安全性要件と攻撃者についてそれぞれ説明する。

（イ）　変換機能による実現方式の分類

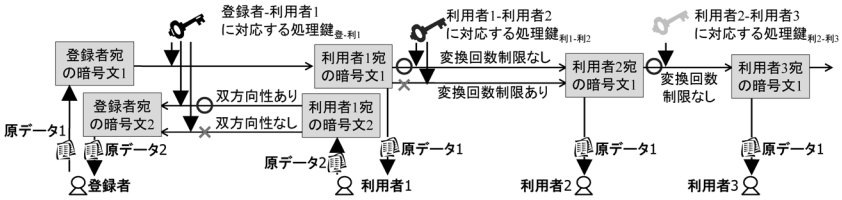
前述の処理フローでは登録者および利用者が各 1 名のみのモデルを想定したが、学界では、登録者および利用者がそれぞれ複数存在する「拡張モデル」が主な検討対象となっている。この拡張モデルでは、複数名の登録者が、登録データを外部サーバにそれぞれ登録すると同時に、利用者として他者が登録した登録データに対する閲覧要求等を行うケースが想定される。学界では、既存方式が実現する変換機能を 2 つの観点から分類していることから、以下では、拡張モデルを前提に各観点を紹介する（図表 13、図表 14）。

21　メンバーからの情報漏えいを防止するには、閲覧終了後にファイルを直ちに削除し、利用者の端末に同ファイルが残存しないようにする仕組みが不可欠である。

図表 13　変換機能による実現方式の分類

		変換回数の制限	
		なし	あり（1 回のみ）
変換の双方向性	なし	※本項目は、該当する研究成果が現在公表されていない。ただし、今後公表される可能性はある。	Ateniese *et al.* [2006], Hohenberger *et al.* [2007], Chow *et al.* [2010], Libert and Vergnaud [2011], Hanaoka *et al.* [2012], Isshiki, Nguyen, and Tanaka [2013]
	あり	Blaze, Bleumer, and Strauss [1998], Canetti and Hohenberger [2007], Matsuda, Nishimaki, and Tanaka [2010][22]	Ateniese, Benson, and Hohenberger [2009]

図表 14　秘匿変換の 2 つの変換機能



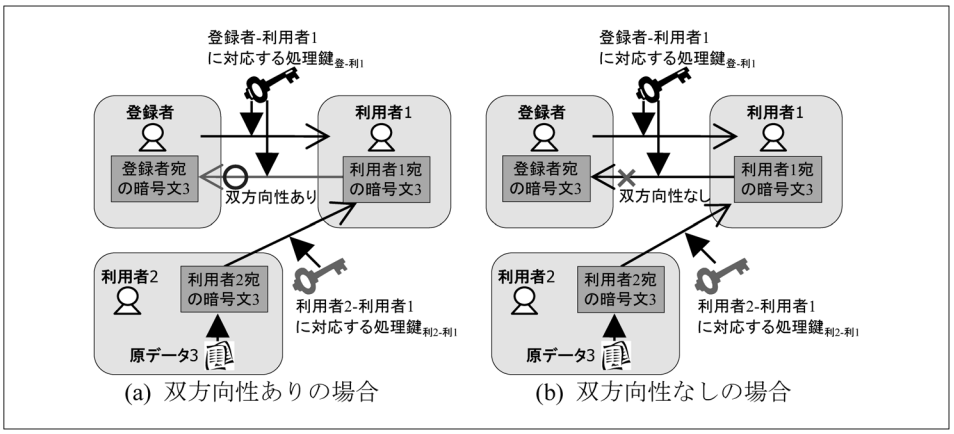
変換の双方向性[23]：「暗号文（図表 14 の暗号文 1）を登録者宛から利用者（同利用者 1）宛に変換する処理」と「暗号文（同暗号文 2）を利用者 1 宛から登録者宛に変換する処理」を、同じ処理鍵を用いて実行できるという性質。同性質を満たす実現方式は、2 種類の変換（登録者宛から利用者 1 宛、およびその逆）を 1 つの処理鍵で実現できるため、利便性の面では優れると考えられるものの、別の利用者（図表 15 の利用者 2）が登録した登録データ（同利用者 2 宛の暗号文 3）を、「登録者に流通させずに利用者 1 のみに流通させるといった流通制御」が困難になるというセキュリティ面での制約が存在する[24]。前述の処理フローで示した処理鍵は、登録者の復号鍵と利用者の暗号化鍵か

22 同方式については、Weng, Zhao, and Hanaoka [2011] により安全性評価に不備があると指摘されたのち、南雲・西巻・田中［2011］により安全性の再評価が行われた。

23 学界では、「変換の双方向性」は「再暗号化鍵（処理鍵）の性質」、「双方向性なし」は「一方向性」、「双方向性あり」は単に「双方向性」とそれぞれ呼ばれている。

24 登録者と利用者 1 の間の変換は双方向性を満たすため、両者に対応した処理鍵があれば、利用者 1 宛の暗号文 3 を登録者宛に変換されてしまう。なお、こうした変換を実行しないように、変換実行に関するポリシーを別途定めておき、同ポリシーを外部サーバに遵守させるという運用も考えられるが、その際、外部サーバが同ポリシーに従うことを別途保証する仕組みが必要になる。双方向性は、拡張モデル（複数の登録者・利用者の存在）を前提としているが、そもそも、同モデルに基づいた現実的な利用シーンが想定し難いという面も否めない。

図表 15　双方向性を満たす実現方式のセキュリティ面の制約



ら生成されているため、同性質を満たさない。この性質を満たすためには、登録者の暗号化鍵／復号鍵と利用者の暗号化鍵／復号鍵をすべて使用して処理鍵を生成する必要があるが、その際、利用者の復号鍵を登録者に漏えいさせずに（逆の場合も同様）、いかに処理鍵を生成するかという課題がある[25]。

変換回数の制限[26]：ある暗号文を登録者宛から利用者（図表 14 の利用者 1）宛に変換後、さらに、別の利用者（同利用者 2）宛の暗号文に変換できるという性質。同性質を満たす実現方式では、登録者宛の暗号文を利用者 1 宛、利用者 2 宛と次々に変換することで、暗号文を転々流通させることができる。逆に、同性質を満たさない実現方式では、登録者が許可した利用者（同利用者 1）以外の利用者（同利用者 2，3 等）に暗号文が流通することを防止できる。ただし、原データを入手した利用者 1 が同データを利用者 2，3 に漏えいするリスクに対して、別途対策が必要になる[27]。

（ロ）　学界で検討対象とされる安全性要件と攻撃者

学界における研究動向をみると、安全性については、原データの保護（安全性要件）を検討対象とした研究が主流となっている[28]。また、検討対象とされる攻撃モ

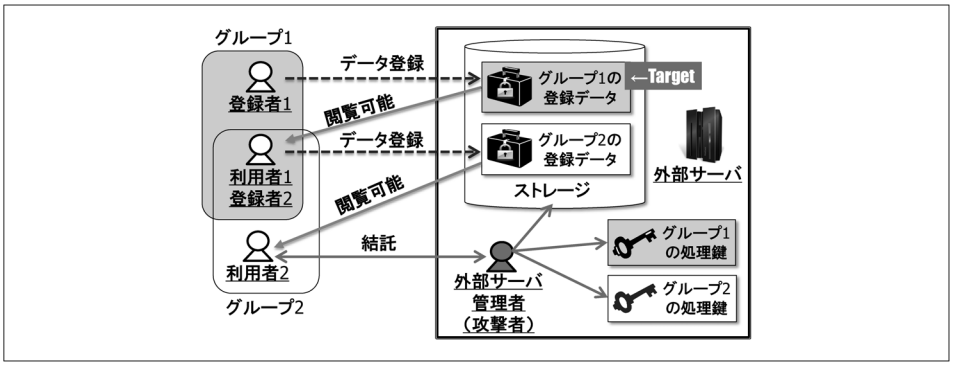
25 信頼できる第三者機関が、登録者と利用者からそれぞれ暗号化鍵／復号鍵を一時的に預かり処理鍵を生成する方法も考えられる。

26 学界では、「変換回数の制限あり」は「single-hop」、「変換回数の制限なし」は「multi-hop」とそれぞれ呼ばれている。

27 前述の代表的な応用例でも触れたが、閲覧したデータ（原データ 1）が利用者 1 の端末に残らないようにする対策（例：端末のシンクライアント化）等もあわせて検討する必要がある。

28 本稿では議論の対象外としたが、外部サーバが適切に秘匿変換を行わなかったり、登録データを改ざんしたりする脅威を想定したうえで、同脅威に対して安全性を確保できる秘匿変換の実現方式が提案されている（川合ほか［2012］、大畑ほか［2013］）。

図表 16　検討対象とされる攻撃モデルの例



備考：利用者 1 は、グループ 1 では利用者として振る舞うものの、グループ 2 では登録者として振る舞う。攻撃者によるグループ 2 の処理鍵の入手を想定しない研究が多いが、それを想定する研究も発表されている（Hanaoka *et al.* [2012]）。

デルをみると、外部サーバに 2 つのグループ（グループ 1, 2）の登録データと処理鍵がそれぞれ保管されており、同サーバの管理者権限を奪取した攻撃者が、グループ 2 の利用者（図表 16 の利用者 2）と結託することで（攻撃者 2）、グループ 1 の登録データに対応する原データを入手できるかという状況を想定する研究が多い。グループが 1 つ（グループ 1 のみ）しか存在しない状況において、攻撃者が同グループの利用者（同利用者 1）と結託した場合、同グループの登録データに対応する原データを入手できることは自明である[29]。

(3)　秘匿計算

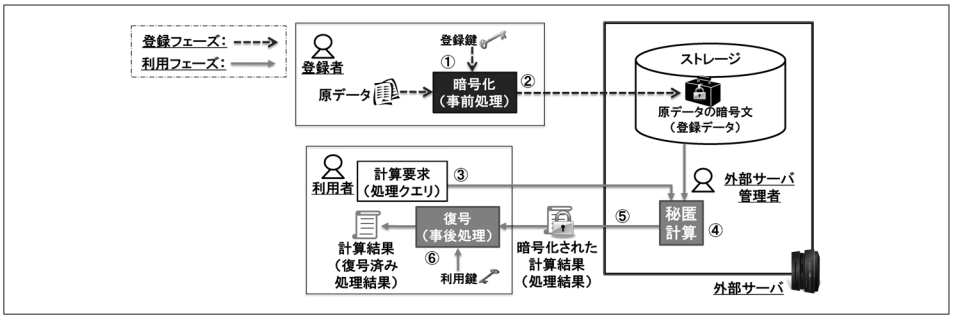
イ．実現アイデアと処理フロー

秘匿計算は、前述のとおり、複数の暗号文（原データの暗号文）に対して、暗号化された状態のまま統計解析等の数値計算を行う技術であり、計算結果も暗号化された状態で得られる。直感的には、暗号文は原データ（平文）を暗号化鍵でマスクしたものとみなすことができ、マスク（暗号化）されたままの状態でも、原データ同士の数値計算が可能な性質（「準同型性[30]」等）を満たす関数（暗号化方式）を用いることにより、秘匿計算を実現する。例えば、平文を e 乗することでマスク（暗

29 グループ 1 の利用者（利用者 1）は、グループ 1 の任意の登録データに対応する原データを閲覧できると考えられるため。なお、攻撃者が利用者 1 と結託し、攻撃者用の処理鍵を偽造するという攻撃を検討対象とする研究もある（Hayashi *et al.* [2011]）。

30 厳密には、暗号化関数を E(·) とし、ある 2 項演算（乗算や加算等）を表す演算子を「∘」としたとき、2 つの平文の暗号文 $E(M_1)$ と $E(M_2)$ の演算結果「$E(M_1) \circ E(M_2)$」が、M_1 と M_2 を乗算した値の暗号

図表 17　秘匿計算の処理フロー



号化）を行う暗号化方式（RSA 等）を利用すると、ある商品の単価の e 乗（P^e）と購入する個数の e 乗（N^e）を掛け合わせることで、合計金額の e 乗（$(P\times N)^e=P^e\times N^e$）をマスクされた状態のままでも計算することができる。以下では、各鍵を生成する処理（鍵生成フェーズ）、原データを外部サーバへ登録する処理（登録フェーズ）、数値計算を行う処理（利用フェーズ）の処理フローを示す（図表 17）。

鍵生成フェーズ：利用者は、暗号化鍵／復号鍵を生成し、復号鍵を利用鍵として保管する。暗号化鍵を登録鍵として登録者に委託する。

登録フェーズ：登録者は、原データを登録鍵で暗号化（事前処理）したうえで（図表 17-①）、このデータを登録データとして外部サーバに保管する（同②）。

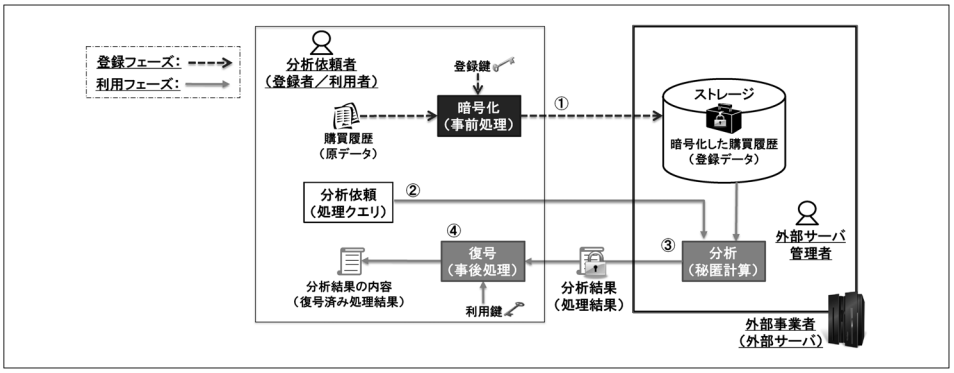
利用フェーズ：利用者は、計算対象の登録データと計算内容を指定した「計算要求（処理クエリ）」を生成し、同サーバに送信する（同③）。なお、秘匿計算では、利用者は事前処理を行わないため、暗号化処理クエリも生成されない。外部サーバは、この計算要求に基づき、登録データに対する数値計算を行い（同④）、暗号化された計算結果（処理結果）を利用者に返す（同⑤）。利用者は、これを利用鍵で復号し、計算結果（復号済み処理結果）を得る（同⑥）。

ロ．代表的な応用例

秘匿計算の代表的な応用例として、データ分析委託サービスに関する例を紹介する（NEC［2013］、日立製作所［2014］等）。同サービスでは、ある企業の従業員（登録者）が、自社の顧客の購買履歴等のデータ（原データ）を外部事業者のサーバ（外部サーバ）に預け、同サーバにデータ解析を委託しつつも、購買履歴等は同サーバに漏えいしたくないという状況を想定している（図表 18）。利用者も同企業

文「$E(M_1\times M_2)$」と等しいとき（すなわち、$E(M_1)\circ E(M_2) = E(M_1\times M_2)$）、同関数は準同型性（「乗法準同型性」）を有するという。なお、加法において同様の関係を満たすときは、「加法準同型」を有するという。

図表 18　秘匿計算のデータ分析委託サービスへの応用例



の従業員である。同企業にとっては、データ分析に必要なサーバを自前で所有する必要がなく、外部事業者のサーバからの情報漏えいも暗号技術的に防止することができるというメリットがある。

ハ．研究動向

（イ）学界において検討対象とされる安全性要件と攻撃者

学界においては、秘匿計算の安全性に関しては、原データの保護（安全性要件）を検討する研究が主流となっている。また、想定する攻撃者に関しては、秘匿検索と同様、理論的には攻撃者が利用者と結託しない場合（攻撃者 1）と結託する場合（攻撃者 2）がありうるものの、利用者が 1 人のみであり結託は行わないという攻撃者（攻撃者 1）を検討対象とした研究が主流となっている。

（ロ）実現可能な演算による分類

1970 年代から、「原データを暗号化したまま計算する」というアイデアが知られていた（Rivest, Adleman, and Dertouzos [1978]）。初期段階では、任意の演算を暗号化したまま可能な方式が実現できておらず、乗算または加算の一方のみを暗号化したまま演算可能な方式（「準同型暗号」と呼ばれる）の実現に留まっていた。こうした方式では、電子投票の集計等の単純な処理を実現できるもの[31]、乗算と加算を組み合わせた複雑な処理（例：統計解析等）を実現することは困難であった。

31 例えば、加法準同型性を有する暗号化関数を E(·) をとし、信任投票において「信任の場合には 1 の暗号文 E(1)」、「不信任の場合には 0 の暗号文 E(0)」を投票することとする。このとき、投票されたすべての暗号文の演算（E(1)∘E(0)∘E(0)∘E(1)∘…）を行うことにより、投票内容を暗号化したまま同投票の集計結果（E(1+0 + 0 + 1 + …)）が得られる。ここで、∘は加算や乗算等の 2 項演算を表す演算子とする。

図表 19　実現可能な演算による分類

実現方式	暗号化したまま実現可能な演算	演算の例	相対的な処理速度
準同型暗号	乗算	$E(M_1)\times E(M_2) = E(M_1\times M_2)$	高速
	加算	$E(M_1)+E(M_2) = E(M_1+M_2)$	
Somewhat 準同型暗号	加算と回数に制限のある乗算の組合せ	$E(M_1)\times E(M_2)+E(M_3)$ $= E(M_1\times M_2+M_3)$	中速 *
完全準同型暗号	乗算と加算の組合せ		低速

備考：E(·) を暗号化関数とし、M_1, M_2, M_3 を平文（原データ）とする。
　　* 乗算の回数が増えるほど処理速度は低下する。

2005 年に漸く、乗算の演算回数に制限があるものの、乗算と加算の両者を暗号化したまま演算可能な方式（「Somewhat 準同型暗号」と呼ばれる。Boneh, Goh, and Nissim [2005]）が提案され、さらに、2009 年にはそうした制限のない方式（「完全準同型暗号」と呼ばれる。Gentry [2009]）が提案された。完全準同型暗号については、処理速度が非常に遅いため[32]、処理の高速化を目的とした研究が始まっている（Gentry, Sahai, and Waters [2013]）。これらの秘匿計算の実現方式をまとめると図表 19 のとおりである。

5.　考察

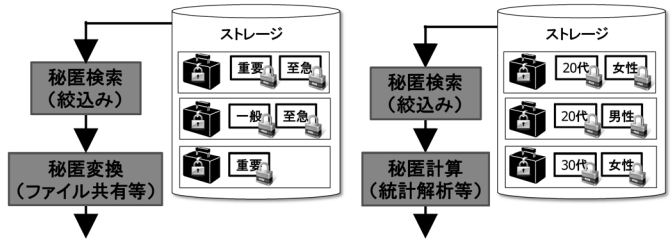
本節では、3 つの暗号化状態処理技術について、組み合わせて利用する方法の可能性、3 つの技術の関係性、ベースとしている暗号技術の安全性の観点からそれぞれ考察を行う。

(1)　組み合わせて利用する方法の可能性

秘匿検索の登録データは、前述のとおり、「暗号化された登録キーワード」と「暗号化された原データ」からなる。利用される暗号技術をみると、登録キーワードに

32 当初の実現方式では、1 回の演算に数十分を要したと言われている（Coron *et al.* [2011]、Gentry and Halevi [2011]）。

図表 20　秘匿検索との組合せの例



(a) 秘匿変換との組合せ　　(b) 秘匿計算との組合せ

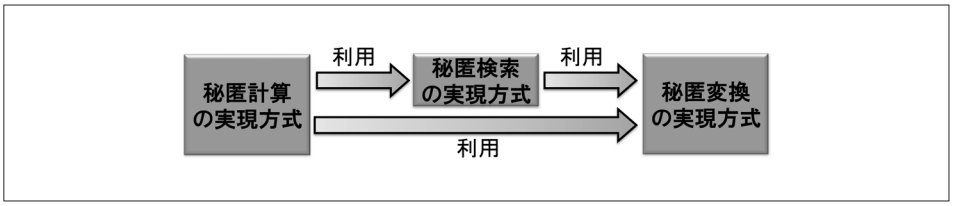
ついては秘匿検索技術が、原データについては通常の暗号技術（AES 等）がそれぞれ利用されている。原データの暗号化については、通常の暗号技術の代わりに秘匿変換技術あるいは秘匿計算技術を用いても秘匿検索には影響を与えないため、これらの技術を組み合わせて適用するという拡張が可能である。こうした拡張を行うことで、秘匿検索のキーワード検索で絞り込んだ原データに対して、ファイル共有のために秘匿変換を行ったり（図表 20(a)）、統計解析のために秘匿計算を行う（図表 20(b)）といった組合せを実現できる。なお、秘匿変換および秘匿計算を実現する既存方式はいずれも原データの暗号化に関わる処理を行うため、両者を組み合わせることは現時点では困難であると考えられる。

(2)　暗号化状態処理技術における 3 つの技術の関係性の活用

3 つの暗号化状態処理技術（秘匿検索、秘匿変換、秘匿計算）は、まったく異なる技術ではなく、一連の関係を有することが近年の研究成果により証明されている[33]。具体的には、3 つの技術のうち、秘匿計算がもっとも基礎となる技術であり、秘匿計算の実現方式を利用することで、秘匿検索や秘匿変換の実現方式を構築可能であることが示されている。また、秘匿検索の方が秘匿変換よりも基礎的な技術で

33 厳密には、①「秘匿計算の実現方式『準同型暗号』を利用すれば、秘匿検索の実現方式『検索可能暗号』を構築可能（小暮［2014］等）」と、②「検索可能暗号を利用すれば、秘匿変換の実現方式『代理人再暗号化方式』を構築可能（Boneh *et al.* [2004]、清藤・中野・四方［2014］等）」という 2 つの研究成果に大別される。上記②については、「検索可能暗号を利用すれば、『ID ベース暗号』を構築可能（Boneh *et al.* [2004] 等）」と「ID ベース暗号を利用すれば、代理人再暗号化方式を構築可能（清藤・中野・四方［2014］）」という研究成果に基づく。なお、ID ベース暗号は、公開鍵暗号であり、公開鍵として任意の文字列（電子メールアドレス等）を用いることが可能。

図表 21　3 つの暗号化状態処理技術の関係性



あり、秘匿検索の実現方式を利用することで、秘匿変換の実現方式を構築可能であることも示されている（図表 21）。

こうした研究成果を活用すると次のようなインプリケーションが得られる。秘匿検索や秘匿変換の実現方式を構築するには、①秘匿計算の実現方式を利用して構築する方法と、②そうした利用を行わずに構築する方法が考えられる。例えば、秘匿検索や秘匿変換を行うシステムを開発する状況を想定した場合、上記の方法①（利用あり）を採用した場合には、実質的には秘匿計算の実現方式を開発するのみでよいため、システム開発の期間が相対的に短縮できると期待される。他方、上記の方法②（利用なし）を採用した場合、一般的には、各技術（秘匿検索、秘匿変換）に最適化した実現方式を構築できるため、方法①よりも処理性能が高くなると期待される。こうした特徴を踏まえると、処理性能に対する要求が高くないシステム（例：システムのプロトタイプ版）の開発には方法①を、処理性能に対する要求が高いシステム（例：本番システム）の開発には方法②を採用することも考えられる。

(3)　ベースとする暗号技術の安全性

暗号化状態処理技術の既存研究や製品をみると、その実現に「ペアリング技術[34]」と呼ばれる暗号の要素技術を利用していることが多い。こうした暗号化状態処理技術の安全性は、「ペアリング技術の安全性」と「個々の実現方式に固有の安全性」の 2 つの観点から評価する必要がある。本稿では、多くの暗号化状態処理技術に共通するペアリング技術の安全性について考察する。

34 ペアリング技術は、「特殊な曲線（楕円曲線）上の 2 点を 1 つの数値に変換する技術」であり、「楕円曲線上の 2 点の加算を 2 つの数値の乗算に変換可能な性質（双線形性）」を有している。また、ペアリング技術を用いた暗号技術を「ペアリング暗号」と呼ぶ。ペアリング技術／暗号の詳細は Blake, Seroussi, and Smart [2005] や CRYPTREC［2008］等を参照。

ペアリング技術の安全性は、「離散対数問題[35]」と「楕円曲線離散対数問題[36]」と呼ばれる 2 つの数学的問題の難しさを根拠としている。こうした問題を暗号分野で利用する場合、同問題を解くことが現実的な時間では困難となるようなパラメータを設定することが推奨される（例えば、解くのに 50 兆年かかるようなパラメータを設定する等。宇根ほか［2010］）。離散対数問題の難しさは、①鍵の長さ（鍵長）に依存しており、鍵長を長くするほど難しくなる。また、楕円曲線離散対数問題の難しさは、②鍵長のほかに、③同問題で利用する特殊な曲線（「楕円曲線」と呼ばれる）の形にも依存している。

このため、安全なペアリング技術を利用するには、上記①～③に注意する必要がある。各問題の鍵長（上記①②）については、米国立標準技術研究所（National Institute of Standards and Technology: NIST）等が 2030 年以降も利用可能な鍵長を公表しており、そうした情報を参照可能である（図表 22）。他方、楕円曲線の形（上記③）については、CRYPTREC [2008] や ISO/IEC [2009] 等の参考情報が存在するが、近年、これまで安全とされていた形の楕円曲線の中にも、そうでないものが含まれることが指摘されており（ellipticnews [2014]、NICT［2012］等）、最新の研究動向をフォローすることが重要になってきている。

図表 22　推奨される鍵長

暗号技術強度指標[37]	推奨鍵長［ビット］		使用推奨期間
	離散対数問題に基づく方式	楕円曲線離散対数問題に基づく方式	
112	2,048	224	2011～30 年
128	3,024	256	2031 年～
256	15,360	512	

資料：NIST [2012]

35 離散対数問題とは、「2 つの数 g と t が与えられたとき、g を s 乗した値が t と等しくなるような自然数 s（すなわち、$t=g^s$ を満たす s）を求める問題」のこと。s が鍵であり、その長さ（ビット長）が鍵長。

36 離散対数問題に、楕円曲線上の点のみで考えるという条件を付けたものが楕円曲線離散対数問題である。具体的には、「楕円曲線上の 2 つの点 G と T から、G を s 倍した点が T と等しくなるような自然数 s（すなわち、T=s×G を満たす s）を求める問題」である。離散対数問題と同様、s が鍵であり、その長さ（ビット長）が鍵長（同問題の詳細等については、清藤・四方［2013］参照）。

37 暗号技術強度指標（「ビット安全性」）は、暗号技術の安全性を評価する際の指針の 1 つである。同指針では、ある暗号技術について、その安全性を破るために必要な計算量が 2^k 回の演算処理に相当する場合、その暗号技術は「k ビット安全」という。

図表 23　本稿で取り上げた暗号化状態処理技術の全体像

暗号化状態処理技術	機能に基づく分類
秘匿検索	・完全／部分一致検索 ・AND ／ OR 検索 ・類似検索
秘匿変換	・変換の双方向性 ・変換回数の制限
秘匿計算	・準同型暗号 ・Somewhat 準同型暗号 ・完全準同型暗号

6.　おわりに

企業等による外部サービスの利用が増加しているのにともない、学界においても、外部サーバからの情報漏えい防止に資する「暗号化状態処理技術」の研究が活発化しており、既に製品として提供されているものも存在する。同技術は、情報漏えいの防止を暗号技術的に保証しており、より高い安全性を求める場合には有望な選択肢であると考えられる。

同技術を利用するに当たっては、まず、どういった処理が可能かを把握する必要がある（図表 23）。また、本稿では検討対象外としたが、外部サーバ管理者が登録データを改ざんしたり、指示通りに暗号化状態処理を行わないリスクを考慮し、同リスクへの対策（電子署名等）の利用も検討することが重要である。

暗号化状態処理技術の研究をみると、喫緊の課題であるにもかかわらず具体的な利用シーンを想像し難いシナリオに基づく研究が存在するほか、同技術のキラーアプリケーションがないといった声も一部の研究者から聞こえてくる。企業等には、自分たちのニーズに合った実現方式を探すというアプローチだけでなく、自分たちの利用シーンや直面している課題等を学界に伝えていくというアプローチも期待したい。こうした企業等のニーズが研究の動機付けとなり、有益な研究成果に結び付いていくと考えられる。

参考文献

宇根正志・黒川貴司・鈴木雅貴・田中秀磨、「暗号ユーザーが暗号アルゴリズムの安全性評価結果をどう活用するか」、『金融研究』第 29 巻第 2 号、日本銀行金融研究所、2010 年、201～228 頁

大畑幸矢・松田隆宏・花岡悟一郎・松浦幹太、「検証可能代理人再暗号化方式の安全性について」、暗号と情報セキュリティシンポジウム、2013 年

川合　豊・松田隆宏・花岡悟一郎・國廣　昇、「代理人再暗号化方式の安全性について」、暗号と情報セキュリティシンポジウム、2012 年

金融情報システムセンター（FISC）、『金融機関等コンピュータシステムの安全対策基準・解説書（第 8 版）』、FISC、2012 年

――――、『平成 24 年度金融機関等のコンピュータシステムに関する安全対策実施状況調査報告書』、FISC、2014 年

小暮　淳・安田雅哉・下山武司・小柴健史・横山和弘、「準同型暗号を用いた秘匿検索」、暗号と情報セキュリティシンポジウム、2014 年

産業技術総合研究所（AIST）、「秘密計算による化合物データベースの検索技術」、産業技術総合研究所、2011 年

情報通信研究機構（NICT）、「次世代暗号の解読で世界記録を達成」、情報通信研究機構、2012 年

清藤武暢・四方順司、「公開鍵暗号を巡る新しい動き：RSA から楕円曲線暗号へ」、『金融研究』、第 32 巻第 3 号、日本銀行金融研究所、2013 年、17～50 頁

――――・中野倫太郎・四方順司、「ID ベース暗号による代理人再暗号化方式の一般的構成法」、暗号と情報セキュリティシンポジウム、2014 年

東芝ソリューション、「安全性を高めた、クラウドサービス向け再暗号化技術を開発」、東芝ソリューション、2011 年

南雲皓介・西巻　陵・田中圭介、「On the Security of the Matsuda-Nishimaki-Tanaka Proxy Re-Encryption Scheme」、暗号と情報セキュリティシンポジウム、2011 年

日立製作所、「クラウド上での情報漏えい防止に貢献する検索可能暗号技術を開発」、日立製作所、2012 年

――――、「暗号化したままデータ分析を行う秘匿分析技術を開発」、日立製作所、2014 年

富士通研究所、「世界初！暗号化したまま統計計算や生体認証等を可能にする準同型暗号の高速化技術を開発」、富士通研究所、2013 年

――――、「暗号化したまま検索が可能な秘匿検索技術を開発」、富士通研究所、2014 年

三菱電機、「『秘匿検索基盤ソフトウエア』を開発」、三菱電機、2013 年

吉田麗生・小田　哲・小林鉄太郎、「部分一致検索可能暗号」、暗号と情報セキュリ

ティシンポジウム、2010 年
王　立華・早稲田篤志・野島　良・盛合志帆、「PRINCESS：プロキシ再暗号化技術を活用したセキュアなストレージシステム」、暗号と情報セキュリティシンポジウム、2014 年
CRYPTREC、「ID ベース暗号に関する調査報告書」、CRYPTREC、2008 年
NEC、「NEC、世界初、データベースの情報を暗号化したまま処理できる秘匿計算技術を開発」、NEC、2013 年
NTT ソフトウェア、「国産初の製品で、クラウドセキュリティ市場に本格参入」、NTT ソフトウェア、2013 年
Abdalla, Michel, Mihir Bellare, Dario Catalano, Eike Kiltz, Tadayoshi Kouno, Tanja Lange, John Malone-Lee, Gregory Neven, Pascal Paillier, and Haixia Shi, “Searchable Encryption Revisited: Consistency Properties, Relation to Anonymous IBE, and Extensions,” Proceedings of CRYPTO, LNCS 3621, Springer-Verlag, 2005, pp. 205–222.
Ateniese, Giuseppe, Karyn Benson, and Susan Hohenberger, “Key-Private Proxy Reencryption,” Proceedings of CT-RSA, LNCS 5473, Springer-Verlag, 2009, pp. 279–294.
———, Kevin Fu, Matthew Green, and Susan Hohenberger, “Improved Proxy Re-Encryption Schemes with Applications to Secure Distributed Storage,” Proceedings of ACM Transaction on Information and System Security, no. 9(1), 2006, pp. 1–30.
Baek, Joonsang, Reihaneh Safavi-Naini, and Willy Susilo, “Public Key Encryption with Keyword Search Revisited,” Proceedings of International Conference on Computational Science and Its Application (ICCSA), LNCS 5027, Springer-Verlag, 2008, pp. 1249–1259.
Bellare, Mihir, Alexandra Boldyreva, and Adam O’Neill, “Deterministic an Efficiently Searchable Encryption,” Proceedings of CRYPTO, LNCS 4622, Springer-Verlag, 2007, pp. 535–552.
Blake, Ian, Gadiel Seroussi, and Nigel Smart, “Advances in Elliptic Curve Cryptography,” London Mathematical Society Lecture Note Series, No. 317, Cambridge University Press, 2005.
Blaze, Matt, Gerrit Bleumer, and Martin Strauss, “Divertible Protocol and Atomic Proxy Cryptography,” Proceedings of EUROCRYPT, LNCS 1403, Springer-Verlag, 1998, pp. 127–144.
Bleichenbacher, Daniel, “Chosen Ciphertext Attacks against Protocols Based on the RSA Encryption Standard PKCS #1,” Proceedings of CRYPTO, LNCS 1462, Springer-Verlag, 1998, pp. 1–12.
Boneh, Dan, Giovanni Di Crescenzo, Rafail Ostrovsky, and Giuseppe Persiano, “Public Key Encryption with Keyword Search,” Proceedings of EUROCRYPT, LNCS 3027,

Springer-Verlag, 2004, pp. 506–522.

———, Eu-Jin Goh, and Kobbi Nissim, “Evaluating 2-DNF Formulas on Ciphertexts,” Proceedings of Theory of Cryptography Conference (TCC), LNCS 3378, Springer-Verlag, 2005, pp. 325–341.

———, and Brent Waters, “Conjunctive, Subset, and Range Queries on Encrypted Data,” Proceedings of Theory of Cryptography Conference (TCC), LNCS 4392, Springer-Verlag, 2007, pp. 535–554.

Canetti, Ran, and Susan Hohenberger, “Chosen-Ciphertext Secure Proxy Re-encryption,” Proceedings of ACM Conference on Computer and Communications Security (CCS), 2007, pp. 185–194.

Chow, Sherman S. M., Jian Weng, Yanjiang Yang, and Robert H. Deng, “Efficient Unidirectional Proxy Re-Encryption,” Proceedings of AFRICACRYPT, LNCS 6055, Springer-Verlag, 2010, pp. 316–332.

Coron, Jean-Sèbastien, Avradip Mandal, David Naccache, and Mehdi Tibouchi, “Fully Homomorphic Encryption over the Integers with Shorter Public-Keys,” Proceedings of CRYPTO, LNCS 6841, Springer-Verlag, 2011, pp. 487–504.

Curtmola, Reza, Juan Garay, Seny Kamara, and Rafail Ostrovsky, “Searchable Symmetric Encryption: Improved Definitions and Efficient Constructions,” Proceedings of ACM Conference on Computer and Communications Security (CCS), 2006, pp. 79–88.

van Dijk, Marten, and Ari Juels, “On the Impossibility of Cryptography Alone for Privacy-Preserving Cloud Computing,” Proceedings of USENIX Workshop on Hot Topics in Security (HotSec), 2010, pp. 1–8.

Dong, Qiuxiang, Zhi Guan, Liang Wu, and Zhong Chen, “Fuzzy Keyword Search over Encrypted Data in the Public Key Setting,” Proceedings of International Conference on Web-Age Information Management (WAIM), LNCS 7923, Springer-Verlag, 2013, pp. 729–740.

ellipticnews, “New discrete logarithm records, and the death of Type 1 pairing,” ellipticnews, 2014.

Gentry, Craig, “Fully Homomorphic Encryption using Ideal Lattices,” Proceedings of ACM Annual Symposium on the Theory of Computing (STOC), 2009, pp. 169–178.

———, and Shai Halevi, “Implementing Gentry’s Fully-Homomorphic Encryption Scheme,” Proceedings of EUROCRYPT, LNCS 6632, Springer-Verlag, 2011, pp.129-148.

———, ———, and Nigel Smart, “Homomorphic Evaluation of the AES Circuit,” Proceedings of CRYPTO, LNCS 7417, Springer-Verlag, 2012, pp. 465–482.

———, Ami Sahai, and Brent Waters, “Homomorphic Encryption from Learning with

Errors: Conceptually-Simpler, Asymptotically-Faster, Attribute-Based," Proceedings of CRYPTO, LNCS 8042, Springer-Verlag, 2013, pp. 75–92.

Goh, Eu-Jin, "Secure Indexes," International Association for Cryptographic Research (IACR) Cryptography ePrint Archive, no. 216, 2003.

Golle, Philippe, Jessica Staddon, and Brent Waters, "Secure Conjunctive Keyword Search Over Encrypted Data," Proceedings of Applied Cryptography and Network Security (ACNS), LNCS 3089, Springer-Verlag, 2004, pp. 31–45.

Hanaoka, Goichiro, Yutaka Kawai, Noboru Kunihiro, Takahiro Matsuda, Jian Weng, Rui Zhang, and Yunlei Zhao, "Generic Construction of Chosen Ciphertext Secure Proxy Re-Encryption," Proceedings of RSA Conference Cryptographers' Track (CT-RSA), LNCS 7178, Springer-Verlag, 2012, pp. 349–364.

Hattori, Mitsuhiro, Takato Hirano, Takashi Ito, Nori Matsuda, Takumi Mori, Yusuke Sasaki, and Kazuo Ohta, "Ciphertext-Policy Delegatable Hidden Vector Encryption and Its Application to Searchable Encryption in Multi-User Setting," IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences, Vol.E96-A(1), 2013, pp. 53–67.

Hayashi, Ryotaro, Tatsuyuki Matsushita, Takuya Yoshida, Yoshihiko Fujii, and Koji Okada, "Unforgeability of Re-Encryption Keys against Collusion Attack in Proxy Re-Encryption," Proceedings of International Workshop on Security (IWSEC), LNCS 7038, Springer-Verlag, 2011, pp. 210–229.

Hohenberger, Susan, Guy N.Rothblum, Abhi Shelat, and Vinod Vaikuntanathan, "Securely Obfuscating Proxy Re-encryption," Proceedings of Theory of Cryptography (TCC), LNCS 4392, Springer-Verlag , 2007, pp. 233–252.

Hwang, Yong Ho, and Pil Joong Lee, "Public Key Encryption with Conjunctive Keyword Search and Its Extension to a Multi-user System," Proceedings of Pairing, LNCS 4575, Springer-Verlag, 2007, pp. 2–22.

International Organization for Standardization (ISO), and International Electrotechnical Commission (IEC), "ISO/IEC 15946-5: Information techniques—Security techniques–Cryptographic techniques based on elliptic curves—Part5: Elliptic curve generation," ISO and IEC, 2009.

Isshiki, Toshiyuki, Manh Ha Nguyen, and Keisuke Tanaka, "Proxy Re-Encryption in a Stronger Security Model Extended from CT-RSA2012," Proceedings of RSA Conference Cryptographers' Track (CT-RSA), LNCS 7779, Springer-Verlag, 2013, pp. 277–292.

Kamara, Seny, and Charalampos Papamanthou, "Parallel and Dynamic Searchable Symmetric Encryption," Proceedings of Financial Cryptography and Data Security (FC), LNCS 7779, Springer-Verlag, 2013, pp. 258–274.

Katz, Jonathan, Amit Sahai, and Brent Waters, "Predicate Encryption Supporting Disjunctions, Polynomial Equations, and Inner Products," *Journal of Cryptology*, 26(2), 2013, pp. 191–224.

Li, Jin, Qian Wang, Cong Wang, Ning Cao, Kui Ren, and Wenjing Lou, "Fuzzy Keyword Search over Encrypted Data in Cloud Computing," Proceedings of IEEE Conference on Computer Communications (INFOCOM), 2010, pp. 441–445.

Libert, Benoît, and Damien Vergnaud, "Unidirectional Chosen-Ciphertext Secure Proxy Re-Encryption," *IEEE Transactions on Information Theory*, 33(3), 2011, pp. 1786–1802.

Matsuda, Toshihide, Ryu Nishimaki, and Keisuke Tanaka, "CCA Proxy Re-Encryption without Bilinear Maps in the Standard Model," Proceedings of Public Key Cryptography (PKC), 2010, LNCS 6056, Springer-Verlag, pp. 261–278.

Moataz, Tarik, and Abdullatif Shikfa, "Boolean Symmetric Searchable Encryption," Proceedings of ACM Symposium on Information, Computer and Communications Security (ASIACCS), 2013, pp. 265–276.

National Institute of Standards and Technology (NIST), "Recommendation on Key Management," Special Publication (SP) 800–57, 2012.

Rivest, Ronald, Leonard Adleman, and Michael L. Dertouzos, "On Data Banks and Privacy Homomorphisms," Foundations of Secure Computation, Academia Press, 1978, pp. 169–177.

———, Adi Shamir, and Leonard Adleman, "A Method for Obtaining Digital Signatures and Public Key Cryptosystems," *Communications of the ACM*, 21(2), 1978, pp. 120–126.

Shen, Emily, Elaine Shi, and Brent Waters, "Predicate Privacy in Encryption Systems," Proceedings of Theory of Cryptography (TCC), LNCS 5444, Springer-Verlag, 2009, pp. 457–473.

Song, Dawn, David Wagner, and Adrian Perrig, "Practical Techniques for Searches on Encrypted Data," Proceedings of IEEE Symposium on Security and Privacy (SP), 2000, pp. 44–55.

Verison Business, "2009 Data Breach Investigations Supplement Report," Verison Business, 2009.

Wang, Cong, Qian Wang, Kui Ren, and Wenjing Lou, "Ensuring Data Storage Security in Cloud Computing," Proceedings of IEEE/ACM International Symposium on Quality of Service (IWQoS), 2009, pp. 1–9.

Weng, Jian, Yunlei Zhao, and Goichiro Hanaoka, "On the Security of a Bidirectional Proxy Re-Encryption Scheme from PKC2010," Proceedings of Public Key Cryptography (PKC), LNCS 6571, Springer-Verlag, 2011, pp. 284–295.

Yoshino, Masayuki, Noboru Kunihiro, Ken Naganuma, and Hisayoshi Sato, "Symmetric

Inner-Product Predicate Encryption Based on Three Groups," Proceedings of Provable Security (ProvSec), LNCS 7496, Springer-Verlag, 2012, pp. 215–234.

補論 1.　暗号化状態処理技術の実用化動向

暗号化状態処理技術について、既に製品化されているものや、今年から 2015 年度にかけて製品としてリリースすることが予定されているものが複数存在する。これらの情報を図表 A-1 に例示する。

図表 A-1　暗号化状態処理技術の製品に関する情報

暗号化状態処理技術	主な用途	処理性能	ベースとなる暗号技術
秘匿検索	ゲノムデータ解析（日立製作所［2012］）	1 万件規模のデータ検索が約 8 ミリ秒で可能。	共通鍵暗号
	ビッグデータ分析（富士通研究所［2013］）	1 万 6 千文字の暗号化データの全文検索が約 1 秒で可能。	格子暗号
	データベース検索（三菱電機［2013］）	10 万件規模のデータ検索が約 1～3 秒で可能。	ペアリング暗号／技術
秘匿変換	ファイル共有（東芝ソリューション［2011］）	1 MB の暗号化データの秘匿変換が約 85 ミリ秒で可能。	
秘匿計算	ビッグデータ分析（日立製作所［2014］）	10 万件規模のデータに対する相関ルール分析が約 10 分で可能。	
	生体認証（富士通研究所［2014］）	生体情報（2,048 次元の生体特徴ベクトル）の照合が約 5 ミリ秒で可能。	格子暗号

備考：プレスリリース等で処理性能を公表している製品・サービスについてのみ記載。

補論 2.　秘匿変換の実現例

秘匿変換を実現する具体的な方法を以下に示す。同方式では、共通鍵暗号方式の 1 つである「ワンタイムパッド」を利用する。同暗号は、平文をビット列とみなし、同じ長さのランダムなビット列（共通鍵）をビットごとに排他的論理和[38] を行うことで暗号化を行う。復号するには、暗号化に用いたランダムなビット列と暗号文の排他的論理和を行えばよい。なお、以下の実現方法では、「登録鍵、利用鍵、処理鍵はそれぞれ 1 回しか利用しない」という条件のもと、外部サーバの管理者権限を有し、他の利用者との結託を行う攻撃者（攻撃者 2）に対して、原データの漏えい防止（安全性要件）を達成できる[39]。

鍵生成フェーズ：登録者および利用者は、互いに乱数（共通鍵）を生成し、これらをそれぞれ登録鍵 K_1 と利用鍵 K_2 とする。そして、登録者と利用者が協力して、登録鍵 K_1 と利用鍵 K_2 から処理鍵 $K_R=K_2 \oplus K_1$ を生成し、外部サーバに預託する。ここで、$\oplus$ は排他的論理和の演算子を表す。

登録フェーズ：登録者は、登録鍵 K_1 を用いて原データ M を暗号化することで、自分宛の暗号文 $C=M \oplus K_1$ を生成し、これを登録データ C として外部サーバに保管する。

利用フェーズ：利用者から登録データ C の閲覧要求を受信した外部サーバは、予め預託されていた処理鍵 K_R を用いて同データに対して秘匿変換を行うことで利用者宛の暗号文 $C'=C \oplus K_R$ を生成し、これを利用者に送信する。利用者は、利用鍵 K_2 を用いて自分宛の暗号文 C' を復号することで、原データ $M=C' \oplus K_2$ を入手する。

38 排他的論理和は、2 進数における 1 桁のビット同士で行う演算の 1 つであり、直感的には「桁上がりを無視した加算」と解釈できる。2 進数においては、1 と 1 を単純に加算すると、桁上がりが発生して計算結果は「10」となるが、排他的論理和では、2 桁目の 1 を捨てて答えを「0」とする。ここで、排他的論理和の演算子を $\oplus$ で表すと、2 つのビットにおいて「$0 \oplus 0=0$」、「$0 \oplus 1=1$」、「$1 \oplus 0=0$」、「$1 \oplus 1=0$」となる。

39 厳密には、同実現例は、攻撃者が予め外部サーバ上に登録されていた暗号文と処理鍵のみ利用可能という前提（「受動的攻撃」と呼ばれる）のもとで安全性要件を満たす。一方、（4 節（2）ハ.）で紹介した多くの方式は、攻撃者が任意に選択した原データの暗号文等を入手可能という、受動的攻撃よりも攻撃者にとって有利な前提（「能動的攻撃」と呼ばれる）のもとでも安全性要件を満たす方式となっている。能動的攻撃は、一見すると非現実的な攻撃に思われるが、暗号技術の実装方法や利用環境等によっては、実際に起こりうる攻撃である（同攻撃が実際に起こりうる状況やその脅威等の詳細については、Bleichenbacher [1998] 参照）。