

CRYPTREC 暗号技術ガイドライン (高機能暗号)

CRYPTREC 暗号技術調査ワーキンググループ
(高機能暗号)

2023 年 3 月

表 1 CRYPTREC 高機能暗号 WG 委員構成

主査	四方 順司	横浜国立大学
委員	岩本 貢	電気通信大学
委員	大原 一真	国立研究開発法人産業技術総合研究所
委員	勝又 秀一	PQShield Ltd. / 国立研究開発法人産業技術総合研究所
委員	金岡 晃	東邦大学
委員	川原 祐人	日本電信電話株式会社
委員	国井 裕樹	セコム株式会社
委員	須賀 祐治	株式会社インターネットイニシアティブ
委員	鈴木 幸太郎	豊橋技術科学大学
委員	花岡 悟一郎	国立研究開発法人産業技術総合研究所
委員	濱田 浩気	日本電信電話株式会社
委員	外園 康智	株式会社野村総合研究所
委員	山田 翔太	国立研究開発法人産業技術総合研究所
委員	米山 一樹	茨城大学
委員	渡邊 洋平	電気通信大学

目次

第 1 章	はじめに	4
第 2 章	高機能暗号技術とその活用法	7
2.1	高機能暗号とは	8
2.2	高機能暗号はどこに使えるか、その有用性	8
2.2.1	CRYPTREC 暗号リストの暗号方式との違い	8
2.2.2	高機能暗号の有用性	9
2.3	高機能暗号の種類と分類	10
2.4	高機能暗号の活用例と標準化動向	10
2.4.1	守秘関連の活用事例と標準化動向	12
2.4.2	認証・署名関連の活用事例と標準化動向	17
2.4.3	その他の高機能暗号の活用事例と標準化動向	21
2.4.4	活用事例からみた高機能暗号の利用方法	28
	参考文献	37
第 3 章	主な高機能暗号技術のアルゴリズム・プロトコルとその性能	46
3.1	守秘を目的とした高機能暗号技術	47
3.1.1	I D ベース暗号	47
3.1.2	属性ベース暗号	49
3.1.3	放送型暗号	53
3.1.4	準同型暗号	55
3.1.5	プロキシ再暗号化	58
	参考文献	60
3.2	認証・署名を目的とした高機能暗号技術	65
3.2.1	属性ベース署名	65
3.2.2	集約 MAC、マルチ MAC、集約署名、マルチ署名	67
3.2.3	グループ署名	70

	3.2.4	リング署名	73
	3.2.5	しきい値署名	76
		参考文献	79
3.3		その他の高機能暗号技術	84
	3.3.1	秘密分散	84
	3.3.2	マルチパーティ計算-秘密分散ベース-	87
	3.3.3	マルチパーティ計算-Garbled Circuit ベース-	90
	3.3.4	ゼロ知識証明	94
	3.3.5	Oblivious Random Access Machine (ORAM)	99
	3.3.6	Private Information Retrieval (PIR)	101
	3.3.7	検索可能暗号	105
		参考文献	108
第 4 章		おわりに	116

第1章

はじめに

ネットワークの高速化、広帯域化、さらには、すさまじい勢いでコンピュータの高性能化に伴い、サイバー空間とフィジカル空間が融合したいわゆる Society5.0 の世界が現実的なものになりつつある。特に、クラウドサービスの充実、人工知能（Artificial Intelligence, AI）技術の進歩等もあいまって、多くの分野でオンラインサービスの需要が高まっている。また、暗号資産の出現により、経済も変化しつつある。さらに、2020年に訪れたコロナ禍においては、すべての対応をオンライン化しなければならない状況が生じ、在宅勤務を含みリモートワークや、リモート会議等のオンラインサービス、それに伴う技術が著しく進歩し、いわゆるニューノーマルな社会を実現するための環境作りが進んできた。

以前より、これらの技術の進歩、社会の変化に伴った現状のサービスの改善や、将来のサービス変化を見越して、既存の暗号技術よりも効率的で高機能な方式の研究開発が進められている。従来技術より、低コスト、高機能の特徴をもつ「高機能暗号」は、ゲノムのような重要な個人情報扱う医療データの活用、AI用の秘密計算、DBの秘匿検索をはじめとするさまざまな用途での利用が期待されている。しかし、どの高機能暗号を選べばよいのか、選ばれた高機能暗号方式は従来の暗号方式よりもどのような点で優れているのか、運用時にはどのようなことに注意することが必要か等、実際に利用する際には専門家以外では、判断基準がわからず、実運用の判断に困難な場合も多い。

CRYPTRECでは、主として電子政府で利用する暗号技術について検討を行っているが、それに加えて、今後さまざまな領域で利用が想定される暗号技術について技術調査を行い、社会に役立つ形で情報提供を行うことを目指している。特に、高機能暗号技術が求められる製品やサービスにおいて、利用者が適切な暗号方式を選択でき、容易に調達できることを目指し、2021年度より CRYPTREC 暗号技術評価委員会の下に、“暗号技術調査ワーキンググループ（高機能暗号）”（以下、

高機能暗号 WG) が設置された。本ガイドラインは、高機能暗号の方式を選択・利用する際の技術的判断に資すること、今後の利用促進を図ることを目的として、高機能暗号 WG が作成したものである。主たる読者として、セキュリティ技術の標準化等を目指すコンソーシアムや団体、情報システムのセキュリティ機能の設計・開発・実装において暗号技術を活用する技術者や、セキュリティ機能を搭載した情報システムの導入を推進する企業等を想定しているが、高機能暗号技術に興味を持つ方に広く読んで頂ければ幸いである。

1 章は、本ガイドラインの総説である。2 章では、高機能暗号の概説をまとめている。まず、対象とする高機能暗号と、その活用例を示し、その上で、高機能暗号を実際に活用する際の手引きを示している。特に、高機能暗号の特徴、代表的なユースケース、方式の選択方法を記載している。3 章では、代表的な高機能暗号のアルゴリズムと性能等を示している。多くの高機能暗号方式が提案されている中、優位な特徴がある方式を取り上げ、アルゴリズムを紹介するとともに、その優位性を示している。

本ガイドラインを読むにあたり、どのような暗号があるかを知りたい読者は、第 1 章、第 2 章までを読み、より暗号技術の詳細内容まで知りたい読者は、第 3 章まで読むことをお勧めする。

本ガイドラインで紹介している高機能暗号技術は、執筆時点までに、主要国際学会で発表されており、有力な攻撃法が発見されておらず、かつ、十分な性能を持つと考えられる方式を選んでいる。このため、本ガイドラインで紹介する高機能暗号は、CRYPTREC 暗号リストの暗号方式とは異なり、CRYPTREC において安全性評価を行った方式ではなく、国際会議等の発表により脆弱性が発見されていないことを安全性の根拠としている。そして、可能な限り最新の情報に基づき安全性や、性能比較、標準化動向等を紹介している。しかしながら、高機能暗号の研究開発は今まさに盛んに行われており、年々新たな方式や評価結果が出ているところであり、記載内容が執筆時点のものであることに留意いただきたい。また、本ガイドラインで紹介する高機能暗号は、その機能の優位な特徴を明確に示す暗号を掲載している。このため、アプリケーションに本ガイドラインに記載されている暗号方式を実装することで、その暗号方式が有する優位な特徴を利用することはできるものの、必ずしもそのアプリケーションにとって最適な方式とは限らないケースもある。この場合、最適な方式を選択するためには、暗号方式の詳細を理解し、改良したり、別の暗号方式を利用する等の検討が必要となることがあることをご留意いただきたい。

本ガイドラインは表 1.1 に示す高機能暗号 WG 委員および CRYPTREC 事務

局で執筆・編集を行った。所属は、WG 委員、事務局を担当した期間におけるものである。また、本ガイドラインを作成する上で、活用事例からみた高機能暗号の利用方法において、日本電気株式会社の横田治樹氏、藤井了氏、三菱電機株式会社の平野貴人氏、川合豊氏、反町亨氏、早坂健一郎氏に多くの有益な情報をご提供いただいた。

表 1.1 高機能暗号ガイドライン執筆者リスト

主査	四方 順司	横浜国立大学
委員	岩本 貢	電気通信大学
委員	大原 一真	国立研究開発法人産業技術総合研究所
委員	勝又 秀一	PQShield Ltd. / 国立研究開発法人産業技術総合研究所
委員	金岡 晃	東邦大学
委員	川原 祐人	日本電信電話株式会社
委員	国井 裕樹	セコム株式会社
委員	須賀 祐治	株式会社インターネットイニシアティブ
委員	鈴木 幸太郎	豊橋技術科学大学
委員	花岡 悟一郎	国立研究開発法人産業技術総合研究所
委員	濱田 浩気	日本電信電話株式会社
委員	外園 康智	株式会社野村総合研究所
委員	山田 翔太	国立研究開発法人産業技術総合研究所
委員	米山 一樹	茨城大学
委員	渡邊 洋平	電気通信大学
事務局	野島 良	国立研究開発法人情報通信研究機構
事務局	青野 良範	国立研究開発法人情報通信研究機構
事務局	伊藤 竜馬	国立研究開発法人情報通信研究機構
事務局	大久保 美也子	国立研究開発法人情報通信研究機構
事務局	金森 祥子	国立研究開発法人情報通信研究機構
事務局	黒川 貴司	国立研究開発法人情報通信研究機構
事務局	篠原 直行	国立研究開発法人情報通信研究機構
事務局	吉田 真紀	国立研究開発法人情報通信研究機構
事務局	小川 一人	国立研究開発法人情報通信研究機構

第 2 章

高機能暗号技術とその活用法

2.1 高機能暗号とは

近年、従来の守秘、認証、署名の機能だけではなく、様々な機能を持つ「高機能暗号」(Advanced Cryptography)の研究開発が進んでおり、学会等で提案されている。高機能暗号は日本が強みを持つ分野であり、日本での研究開発が盛んである。そして、日本発の高機能暗号が、多くの著名な国際会議で発表されている。また、ISO/IEC 等において標準化が進められている高機能暗号技術もある。

様々な機能を有する高機能暗号技術は、今後の発展が予想される 5G および Beyond5G の社会における種々のサービスで利用できる可能性があり、Internet of Things (IoT) や AI にとどまらず、Society5.0 における基盤となる Cyber Physical System (CPS) といった次世代のネットワークサービスを構築する上でも有効なセキュリティ技術の一つとなることが期待されている。

一方で、これまで提案されてきた高機能暗号技術には、様々な機能を有する方式が存在する。機能に応じて、それぞれ暗号技術が発展しており、それらを一つの指標で比較することは困難である。このため、国際的に「高機能暗号」に対して一般的に合意されている定義はない。ただし、ガイドラインを読み進めるにあたり、高機能暗号を定義しておくことは読者の理解の支えになると考えられる。以上の状況を鑑み、本ガイドラインでは、

「従来の暗号技術に対して、機能が追加・向上される等の優位性を主張する暗号技術、および、従来の暗号技術では困難であった事象を解決できる等の新規機能を有することを主張する暗号技術」

をスコープとし、従来の暗号技術に対して、なんらかの機能・性能で優位性（高機能性）を持つ暗号技術を高機能暗号における主な対象とする。ここで述べた従来の暗号技術とは、CRYPTREC 暗号リスト [1] に挙げられているような暗号基礎技術を指す。そして、この高機能性については、各項目において明示することにする。

2.2 高機能暗号はどこに使えるか、その有用性

2.2.1 CRYPTREC 暗号リストの暗号方式との違い

CRYPTREC 暗号リスト [1] には、公開鍵暗号、共通鍵暗号、ハッシュ関数、暗号利用モード、メッセージ認証コード、認証暗号、エンティティ認証が記載されて

いる。さらに、公開鍵暗号は、署名、守秘、鍵共有に分かれている。また、共通鍵暗号は、64 ビットブロック暗号、128 ビットブロック暗号、ストリーム暗号に分かれ、暗号利用モードは、秘匿モード、認証付き秘匿モードに分かれている。

本ガイドラインで紹介する高機能暗号方式は、これらの暗号方式単独では実現できない機能を持つ暗号方式、もしくは、これらの暗号方式で実現するには非効率である事象を効率的に実現する暗号方式である。前者の例としては、CRYPTREC 暗号リストにおける公開鍵暗号では鍵生成においてランダムに出力される値が公開鍵として使用される。これに対し、任意の記号列、例えば、人名やメールアドレス、を公開鍵として利用することを可能とした高機能暗号が開発されている(第 3.1.1 章 ID ベース暗号参照)。また、後者の例としては、一つの平文を複数の利用者に送る際に、復号権限のある利用者の人数分の暗号文を生成し全利用者に暗号文を送ることに比べて、さらにコンパクトな暗号文のサイズで同じ機能を達成する高機能暗号も開発されている(第 3.1.3 章放送型暗号参照)。

第 1 章でも述べたように、本ガイドラインで紹介する高機能暗号方式は、CRYPTREC により安全性評価を行った方式ではない。これは、高機能暗号は発展途上にあるが、できる限り最新の情報を提供することを目指したためである。発展途上にある最新の技術を CRYPTREC で評価をするためには、多大な時間を要することになり、逆に、最新の技術が掲載できなくなる。とはいえ、ガイドラインに掲載するためには、安全性が保障されない技術を記載するわけにはゆかない。そこで、主要な国際会議等の発表により脆弱性が発見されていないことを安全性の根拠とし、世界の研究者を評価者と考え、安全性を担保する方針とした。

2.2.2 高機能暗号の有用性

本ガイドラインで扱う高機能暗号は、従来の暗号技術と比べ、

- ランダムな暗号化鍵ではなく、良く知っているメールアドレスや人名を暗号化鍵に利用できる。
- 属性、例えば課長以上の役職の職員、を指定した、ファイル、ドキュメントへのアクセス制御ができる。
- 復号鍵の漏洩対策として、複数人の復号情報が集まらなければ、もとのデータに復号できない設定にできる。
- 暗号化したまま論理演算や算術演算ができる。
- 個人情報と秘匿したまま、あるグループに所属しているメンバーシップを認

証できる。

- 個人が実際に所有する秘密情報を公開することなく、その秘密を持っていることを他者に証明することができる。
- DB へのアクセスパターンを秘匿することができる。
- 管理者等によるコンピュータの RAM の解析を防ぐことができる。
- DB 管理者等に、どんなデータを検索したかを秘匿したまま、所望のデータを検索できる。

等の特徴を持つ暗号技術であり、これらの特徴が従来の暗号技術よりも優位な点となる。

このように様々な高機能暗号があるが、その優位性は種別毎に異なっている。このため、高機能暗号を使用するにあたり、その目的にあった高機能暗号の選択が必要となる。

2.3 高機能暗号の種類と分類

2.2.2 章で述べたように、高機能暗号は、その使用目的にあった選択が必要となる。そこで、本ガイドラインでは、高機能暗号の目的に応じて、“守秘”、“認証・署名”、“その他”の3つに分類した。

守秘は、第3者に秘匿する文書等を暗号化するための技術である。認証・署名は、文書等を作成する人、文書等を送る人、および、通信している機器の認証を行う、もしくは、それらに署名を付与するための技術である。その他については、上記の守秘、認証・署名には属さない新たな技術である。例えば、複数の人やサーバが協調しつつ各利用者が持つ秘密情報は秘匿して計算を行う方式や、秘密情報を提示することなく個人が秘密情報を持っていることを提示する方法等の技術である。

守秘と認証・署名は CRYPTREC 暗号リストにも同じ分類があるが、その他については、CRYPTREC 暗号リストの分類では、どの分類にも当てはまらない暗号技術となる。表 2.1 に、本ガイドラインで扱う暗号方式の一覧を示す。

2.4 高機能暗号の活用例と標準化動向

以前は、セキュリティ技術は付加機能であり、開発コストが高く、敬遠され、進んで導入する組織は多くなかった。現在では、企業からの情報漏洩は、企業のイメージを落とすことにもつながるため、たとえ一部の情報であっても漏洩することがあってはならない状況である。そして、サイバー攻撃等による情報漏洩の危

表 2.1 本ガイドラインで扱う高機能暗号一覧

分類	高機能暗号	特徴
守秘	I D ベース暗号	任意の記号列を公開鍵にすることが可能
	属性ベース暗号	受信者の属性に応じた復号権限の付与が可能
	放送型暗号	多数の受信者に対して放送用のコンテンツの効率的な復号権限の制御が可能
	準同型暗号	暗号化した状態での演算が可能
	プロキシ再暗号化	サーバにおいて暗号化鍵を変更することが可能
認証・署名	属性ベース署名	個人ではなく、属性に応じた署名権限の付与が可能
	集約 MAC、マルチ MAC、集約署名、マルチ署名	複数の MAC / 署名をコンパクトに圧縮することが可能
	グループ署名、リング署名	グループを構成するメンバの誰かが匿名で署名をすることが可能
	しきい値署名	一定数以上の署名者がそろうことで、一つの署名生成が可能
その他	秘密分散	秘密データを複数に分割し秘匿する方法
	マルチパーティ計算	複数のサーバが入力情報を秘匿したまま協調的に計算して結果を得ることが可能
	ゼロ知識証明	証明者が検証者に命題の正しさを証明でき、それ以外の情報を漏らさないことが可能
	Oblivious Random Access Machine (ORAM)	サーバ内のデータアクセスにおいて、データの秘匿に加えてデータへのアクセスパターンをサーバに対して秘匿することが可能
	Private Information Retrieval (PIR)	サーバ内のデータアクセスにおいて、どのデータにアクセスしたかをサーバに秘匿することが可能
	検索可能暗号	サーバ内のデータへのキーワード検索において、検索キーワードを秘匿したままデータ検索が可能

険性もあり、セキュリティ対策を堅実に行うことが組織のアピールにもつながり、セキュリティ技術を積極的に導入する組織が増加している。

さらに、社会環境も変化し、種々多様なサービスが発展してきている。この変化に伴い、ネットワークや、データの保護方法も変化し、高機能暗号が期待され、提案されている。また、すでに導入、活用され、実用化の段階に入っている高機能暗号も増加している。以下では、個々の高機能暗号に対し、どのような活用例が

考えられているかを示すとともに、使用可能なライブラリや、標準化に向けた活動がある方式については標準化動向を示す。その後、活用事例に対し、高機能暗号を利用した実例を紹介する。

2.4.1 守秘関連の活用事例と標準化動向

■ID ベース暗号 ID ベース暗号は、利用者の ID をはじめとする任意の文字列を公開鍵に利用できる公開鍵暗号方式の一種である。暗号文を受け取る相手の名前、メールアドレス、ID、企業名、部署名等、よく知っており、一目で相手ができる情報を公開鍵として利用することができるため、相手とは全く無関係の公開鍵を利用した従来の暗号方式よりも利便性に優れている。

具体的には、メールの本文や添付ファイルを暗号化するサービス等が商用化されている。そのメールサービスでは、個々の利用者の公開情報であるメールアドレスを公開鍵として、それに対する秘密鍵がサービス事業者により生成され、秘密鍵はそれぞれの利用者に送られ保持される。メールの送信者は、メール送信先のメールアドレスを公開鍵（暗号鍵）として、メール本文を暗号化する。メール受信者は、メールアドレスに対する秘密鍵（復号鍵）を持っているので、このメールを復号して読むことができる。秘密鍵は、メールアドレスの受信者しか持っていない情報であるため、メールの守秘性が保たれる。この方式では、送信者はメール受信者の公開鍵を知る必要はなく、メールアドレスだけを知っていれば、受信者が復号できる暗号文を作ることができるため、従来の公開鍵暗号のように受信者の公開鍵を取得するための通信等の必要がない。

ID ベース暗号では、複数のライブラリがオープンソースソフトウェアとして公開されるなどソフトウェアが充実しており、パフォーマンスも高い [2]。まず、ID ベース暗号に限らず近年の多くの公開鍵暗号方式で使われている Pairing と呼ばれる基盤技術がある。この Pairing をベースに最初に作られたライブラリが PBC ライブラリ (Pairing-based Cryptography Library) [3] である。PBC ライブラリの中には ID ベース暗号関連のソフトウェアも含まれている。PBC ライブラリ以外にも ID ベース暗号関係のライブラリは存在するが、PBC ライブラリが最も参照され、使用されている。商用のサービスとしては Voltage 社による電子メールの暗号化ソリューションである Voltage SecureMail がある [4]。また Cloudflare 社が提供する CDN 顧客の秘密鍵管理のソリューションである Geo Key Manager において ID ベース暗号の発展である Identity-based Broadcast Encryption と Identity-based Revocation が採用されている [5]。

また、標準化活動も活発である。IETF では 2007 年 12 月に RFC 5091 として

Boneh-Franklin の手法と Boneh-Boyen の手法が標準化された [6] ことを皮切りに、ID ベース暗号関連の標準化がすすめられた。RFC 5091 をもとにデータ構造等が RFC 5408、5409 として 2009 年 1 月に標準化された [7, 8]。2011 年に入り、ID ベースの認証鍵交換 (ID based Authenticated Key Exchange, IDAKE) 手法が RFC 6267 として標準化され [9]、2012 年に、IDAKE 自体が RFC 6539 [10]、ID ベース暗号向けの Certificateless Signature 手法が RFC 6507 [11]、境-笠原の手法が RFC 6508 として標準化された [12]。IEEE では P1363.3 として ID ベース暗号だけではなくより広い Identity-based Cryptography (IBC) として標準化の活動が行われ、2013 年に IEEE Std 1363.3-2013 として標準化されている [13]。そこでは境-笠原の Key Encapsulation Mechanism (KEM)、Boneh-Boyen の KEM、Boneh-Franklin と Boneh-Boyen の ID ベース暗号方式が Identity-based encryption scheme として標準化されている。ISO/IEC でも ID ベース暗号だけではなくより広い範囲で Identity-based ciphers として ISO/IEC 18033-5:2015 が策定されている。そこでは IEEE Std 1363.3 と同様に Boneh-Franklin、Boneh-Boyen、境-笠原の手法が標準仕様として掲載されている [14]。

ID ベース暗号の活用に関する学術の研究としては、2003 年に ID ベース暗号のパラメータ配付に DNS を利用する方式が Smetters らにより提案されている [15]。また 2013 年に Ruoti らによる電子メール暗号化の高いユーザビリティを実現する Private Webmail (PWM) において ID ベース暗号が採用されている [16, 17]。Miklejohn らは交通系の料金収受システムの提案内において Blind Identity-based Encryption を利用し、被発行者 ID を利用した暗号化を行っている [18]。機器認証において ID ベース暗号を用いる手法は Software Defined Network (SDN) の研究において 2012 年の Veltri らの研究を筆頭に、2010 年代半ばに複数研究されている [19, 20, 21]。

■属性ベース暗号 属性ベース暗号は、暗号文の復号の際に、暗号文の受信者の肩書、会社名、地域名等の属性に応じて復号権限を変更することができる公開鍵暗号方式の一種である。より正確には、暗号文または復号鍵（秘密鍵）に属性に対応したアクセス構造を持たせることにより、当該暗号文の復号権限を一括で制御できる。従来の公開鍵暗号方式により属性毎のアクセス構造を持たせる場合、同じ属性を持つ利用者の公開鍵を用いて、利用者数に応じた暗号文を作ることになる。これに対し、属性ベース暗号では、暗号文は一つでアクセス構造を持たせることができ、効率性において従来の公開鍵暗号方式よりも優れている。

属性ベース暗号の活用事例としては、クラウド環境を介したファイル交換サービスにおける暗号化されたファイルへのアクセス制御の仕組みをセキュアかつ効

率よく実現するために属性ベース暗号を利用しているソリューションが提供されている [22]。

具体的には、ある企業内のファイルにおいて、部長、課長は閲覧可能であるが、他の職員は閲覧できないようにするソリューションである。そのソリューションでは、部長、課長が持つ秘密鍵には、部長、課長に対応する属性情報が埋め込まれている。他の職員には、部長、課長の属性情報が埋め込まれていない秘密鍵が配布されている。ある企業内の機密ファイルに、部長、課長の属性をもつ復号鍵であれば復号可能となるように、公開鍵（暗号化鍵）と部長、課長という属性を用いて暗号化し、クラウドに保存する。クラウドからファイルの取得は誰でも可能であるが、復号時に、復号者の秘密鍵に埋め込まれた属性が適合しなければ復号できない。すなわち、属性に応じた復号のアクセス制御が可能となる。

一般に、公開鍵暗号においては、暗号文生成時における復号権限をもつエンティティは唯一に指定されるため、複数エンティティに復号権限を与える場合にはその数に比例する暗号文の生成が必要である。一方、属性ベース暗号では上記の機能を単一の暗号文で実現可能となる。属性ベース暗号は、暗号文ポリシー型と鍵ポリシー型に大別され、それぞれ暗号文と秘密鍵にアクセス構造を埋め込む技術である。利用者は、この特徴に応じて、どちらかの型を選択することになる。すなわち、ユースケースにおける暗号文と秘密鍵の更新頻度を比較し、相対的に更新頻度が少ない方へアクセス制御を埋め込む型を選択する方が効果的である。

そして、属性ベース暗号の様々な実現手法を実装したライブラリがオープンソースソフトウェアとして公開されている [23, 24, 25]。[23] は、ID ベース暗号の項で述べた PBC ライブラリを用いて作成された暗号文ポリシー型属性ベース暗号のソフトウェアである。[24] は Garg らによる GCH+13 方式 [26] の属性ベース暗号を用いて作成された Java ベースのライブラリである。[25] は Goyal らの属性ベース暗号 [27] をもとにして作成されたライブラリである。

■放送型暗号 放送型暗号とは、一つの暗号化鍵を用いて暗号化されたコンテンツを、複数の利用者が同時に復号できるようにする方式であり、暗号化鍵と復号鍵が異なる公開鍵型放送型暗号と、暗号化鍵と復号鍵が等しい共通鍵型放送型暗号がある。複数の利用者に同じコンテンツを送る場合であっても、暗号化鍵は一つしかなく、すなわち、暗号化は一度でよい、という点で効率性に優れている。

公開鍵型放送型暗号は、利用者毎に異なる復号鍵を持つが、暗号化のための公開鍵は一つである。暗号化鍵は公開情報であり、公開情報のみで誰でも暗号化が可能だが、この性質を積極的に活用可能な応用先が乏しく、現時点では具体的な社会実装例は見当たらない。

共通鍵型放送型暗号は、暗号化鍵と復号鍵が同一であり、また、利用者すべてが同じ復号鍵を持つ方式である。誰でも暗号化を行うことができる公開鍵暗号型放送型暗号とは異なり、暗号化を行うことができる機関が暗号化鍵を共有したエンティティに限定されるものの、大規模コンテンツ配信システム等においては、暗号化するエンティティがごく少数であるため、そのような制約はそれほど問題とはならず、大規模コンテンツ配信の実用システム上では共通鍵型放送型暗号方式が利用されている。また、Blu-ray 等のメディア用にも使用されており、具体的には、Blu-ray 等のコンテンツ保護に用いられる Advanced Access Content System (AACS) [28] は、Naor らにより開発された放送型暗号である SD 法 [29] と共通鍵暗号を組み合わせた共通鍵型放送型暗号に基づき構成されている。AACS については、デジタル家電業界、パソコン業界、映画業界の主導により、規格策定団体 AACS LA [30] が設立され、IBM、Intel、Microsoft、Panasonic、SONY、Disney、Toshiba、Warner Brothers 等により運営されている。

■**準同型暗号** 準同型暗号は、データを暗号化したまま演算処理（四則演算等）ができる暗号技術である。可能な演算は暗号方式により異なっており、加算のみ、乗算のみ、加算と乗算の両方が可能な方式等がある。暗号化状態での演算であるため、演算の途中でデータが漏洩したとしても、生のデータが解読されることはなく、データを処理速度の速いコンピュータに預けて演算を行う分析等の応用例が具体化されている。具体的には、機械学習の分野で準同型暗号を活用することにより、学習データのプライバシーを確保しつつ機械学習モデルを生成する手法に関する研究開発等がある。

すでに実証実験に至っている事例もある。複数の金融機関の取引データを利用して不正取引検知の精度を向上することを目的とする実証実験である。この実験では、複数の金融機関が持つ取引データを持ち寄り、不正取引検知用の AI モデルを構築する方法が提案されている [31]。複数の金融機関のデータは、他の金融機関や中央サーバには秘匿すべきデータとして扱う。そして、それらを AI の連合学習に利用する。深層学習を行う際には、DeepProtect と呼ばれるプライバシー保護深層学習技術が使われている [32, 33]。DeepProtect では、準同型暗号により学習中のパラメータ（勾配情報）を暗号化して中央サーバに送り、中央サーバでは、暗号化したまま、学習モデルのパラメータの更新を行う。

鉄道会社の IC カードの分析に秘密計算を取り入れる検証等も実施されている [34]。そして、IC カードの利用履歴、乗降履歴や購買履歴等のセキュアな利用に取り組んでいる。この検証では、加算型の準同型暗号の一つである Paillier 暗号 [35] が使用されている。また、暗号化前と暗号化後の順番が変化しない順序保

存型暗号（Order Preserving Encryption）も活用しているとされている。

準同型暗号については、ライブラリ、ソフトウェアが数多くオープンソースソフトウェアとして公開されている。Brakerski らによる準同型暗号 BGV12 方式 [36] と Cheon らによる準同型暗号 CKKS17 方式 [37] を基に開発されたライブラリ HElib [38]、Brakerski による準同型暗号 Bra12 方式 [39]、Fan らによる準同型暗号 FV12 方式 [40]、CKKS17 方式に基づくライブラリ SEAL [41]、CKKS17 方式に基づくライブラリ HEAAN [42]、Bra12 方式、FV12 方式、BGV12 方式、CKKS17 方式、Ducas らによる準同型暗号 DM15 方式 [43]、Chillotti らによる準同型暗号 CGGI16 方式 [44] に基づくライブラリ PALISADE [45]、Bra12 方式、FV12 方式、CKKS17 方式に基づくライブラリ Lattigo [46] 等がある。さらに、HEAAN を用いたプラットフォーム等の製品 [47] が商用化されている。

準同型暗号の標準化も行われており、標準化団体 Homomorphic Encryption Standardization [48] は準同型暗号のセキュリティ、API & アプリケーションの標準化を目指している。この標準化では、記法と準同型暗号化のセキュリティプロパティ、完全準同型暗号スキーム、推奨パラメータ値、標準的な攻撃方法とその耐性、推定実行時間だけでなく、将来を見越した耐量子計算機対応としてラティス攻撃、LWE に対する Arora-Ge 攻撃、RLWE に対する代数攻撃方法とその耐性が標準化の対象となり、今後も、API 設計、アプリケーション例が追加される予定となっている。さらに、ISO/IEC では、単一演算型の準同型のアルゴリズム（Exponential ElGamal encryption、Paillier encryption）の暗号処理にかかるプロセスを規定している [49]。

■**プロキシ再暗号化** プロキシ再暗号化は、ある暗号文を復号せずに、暗号文の作成者とは異なるエンティティ（プロキシ）が、平文が同じである別の暗号文に変換する暗号方式である。入力された暗号文と出力される暗号文では、秘密鍵が異なることが一般的であり、このため、秘密鍵の更新が必要となるアプリケーションや、所有者変更等のアプリケーションでの応用例が報告されている。

例えば、復号を実行せずに動的に受信者を変更できるため、動的にアクセス権限の変更をすることを想定するクラウドストレージ等において特に有用であると考えられる。そして、NuCypher 社は、Umbral PRE と呼ばれるプロキシ再暗号化方式を開発し、これを用いた事業展開を進めている [50]。ここでは、サーバにおかれた暗号化データを利用者に配布する場合、その利用者の秘密鍵で復号できるようにアクセス制御する方式等が提供されている。また、東芝も独自に開発したプロキシ再暗号化方式を用いて、デジタル貸金庫と呼ばれるサービスを 2021 年まで展開していた [51]。デジタル貸金庫では、サーバに蓄えられたデータを、複数の

デバイスで利用できるようになっている。ただし、個々のデバイスの復号鍵は異なっている。この個々のデバイスで利用するために、サーバに蓄えられた暗号文から、利用するデバイスの復号鍵で復号できる暗号文にプロキシ再暗号化を利用して変換していた。

2.4.2 認証・署名関連の活用事例と標準化動向

■属性ベース署名 属性ベース署名とは、属性ベース暗号の署名版と考えて差支えない。属性ベース暗号の場合は、復号する利用者の属性に応じて、より正確には、復号する利用者が持つ復号鍵に応じて、暗号文へのアクセス制御を行っている。これに対し、属性ベース署名では、署名者の持つ署名鍵が、署名者の属性に応じて異なっており署名者の属性集合が埋め込まれた署名鍵になっている。そして、ある文書に対して署名鍵を用いて署名を生成するわけであるが、その署名の中に属性に関するアクセス制御構造が埋め込まれる。署名検証において、署名者の持つ属性集合が、アクセス制御のために埋め込まれた構造にマッチしている場合は、署名検証をパスすることができる。すなわち、署名者の属性に応じて、検証判定ができる署名方式である。

この属性ベース署名を使うことで、社内に送付する文書において、名前ではなく、文書作成者の属性（役職、部局等）等に応じた署名を付与することができる。ある社の A さんが作成した文書を、その課長、部長等が押印して回付することはよくあることである。紙媒体の場合、A さんの印、課長印、部長印が押印される。この課長印、部長印は、課長もしくは部長の権限を持つ人が、課長／部長の責任に応じた役割を果たし、文書を確認したことを、文書の受領者に示すものであり、課長、部長の名前よりも、その役職に意味がある印である。さらに、詳細に、総務部の課長印のように、部署名も必要なこともある。この押印を電子的に行う場合、属性ベース署名が役に立つ。

総務部の課長 B さんは、” 所属：総務部”、” 役職：課長” の属性を持つ。このため、B さんの署名鍵（秘密鍵）は、” 所属：総務部”、” 役職：課長” に応じた署名鍵となる。B さんが署名を生成する際に、署名を受理するためのアクセス条件を付与する。この例の場合、” 総務部” かつ ” 課長” であれば、受理するという条件になる。文書の受信者は、署名がアクセス条件にあっていかどうかを確認することになる。B さんの署名は、” 総務部” かつ ” 課長” の属性であるため、署名検証が通る。もし ” 総務部” かつ ” 課長” ではない第 3 者が署名を偽造したとしても、署名検証でエラーとなる。

この例で示す通り、名前による確認ではなく、役職や所属部署名による確認を

行うことができる。もちろん、企業によっては、課長印、部長印以外に、名前が必要な場合もある。この場合は、”名前：Bさん”または”名前：Cさん”または”名前：Dさん”等をアクセス条件に入れることで対応ができる。

また、属性の作り方を工夫することで、多くの活用方法が考えられ、例えば、ブロックチェーンやユーザ認証において匿名性を確保するために属性ベース署名を活用する手法の研究が行われている [52, 53]。

■集約 MAC・マルチ MAC・集約署名・マルチ署名 集約 MAC は、異なる複数の文書に対し、複数もしくは一つの秘密鍵でメッセージ認証符号 (Message Authentication Code, MAC) を付与する場合を効率よくするための技術である。例えば、企業内で文書に対し、文書の作成者とその上司、および、別の部の担当者等、複数の人が異なる秘密鍵により MAC を付与する場合である。通常の MAC 方式であれば、MAC を付与する人数が多くなると、人数に比例して MAC 部分のサイズが大きくなる。これに対し、集約 MAC では、複数の MAC を単一の MAC に集約するため、MAC サイズが小さくなるという優位性を持つ。

マルチ MAC は一つの文書に対し、複数の異なる秘密鍵で生成された MAC を統合し、集約する技術である。MAC を付与する文書が共通であり、集約 MAC の特殊ケースとして捉えることができる。そして、集約 MAC と同じように、MAC サイズが小さくなる優位性を持つ。

集約署名は集約 MAC の電子署名版と考えることができる。複数の文書に対し、複数人が署名するような場合を効率よくするための技術である。通常の署名方式であれば、署名を付与する人数が多くなると、人数に比例して署名部分のサイズが大きくなる。これに対し、集約署名では、複数の署名を単一の署名に集約するため、署名サイズが小さくなる、という優位性を持つ。また、署名を検証する人は、これを効率よく検証できる提案もある。すなわち、集約された署名の人数分の検証を行うのではなく、一回の検証処理で行う方式である。

マルチ署名は、一つの文書に対し、複数人が署名を行う場合、この署名を単一の署名にまとめる技術である。集約署名と同じように、署名サイズが小さくなる点がメリットである。

集約 MAC および集約署名については、現時点で具体的なサービス等の提供は行われていないが、ITU により廣瀬らによる方式 [54]、佐藤らによる方式 [55] の技術内容の標準化が行われている [56]。また、この技術を用いた認証関連のデータ量の削減効果を考慮し、認証に係るインターネット上のトラフィック量削減を目的とする活用に関する研究が行われている [57]。

マルチ署名は、暗号資産 Bitcoin やブロックチェーンプラットフォームである

Ethereum におけるマルチシグウォレットにおいて活用されている。マルチシグウォレットは、複数のユーザが共同利用することを前提とし、取引の実行には全てのユーザの同意が必要となるものであり、同意形成の仕組みとしてマルチ署名が利用されている。例えば、信頼できる第三者に署名生成鍵を委託することにより、他のユーザによる不正送金等を防ぐことを目的に利用される [58, 59]。

■**グループ署名** グループ署名は、署名者があるグループの一員であることは明示的に分かるが、グループ内の誰であるかまでは分からない匿名性を保証することができる署名方式である。一般の署名方式では不可能であった、署名者のプライバシー保護、匿名性が必要となるアプリケーションにおいて需要がある。

例えば、プログラム開発での確認作業、いわゆる Attestation において、2000 年代からグループ署名が利用されるようになった。ここでの Attestation とはプログラムが製造者やユーザの意図した通りに動作しているかを確認するための処理であり、ブート結果や周辺機器の接続状況等の予め想定された処理結果に対してデジタル署名をつけて正当性を確認できるようにするものである。集中管理するケースにおいては通常のデジタル署名や PKI を利用した場合、Attestation をしたエンティティの活動に対するプライバシー侵害が起こりうるため、グループ署名を利用してそれを回避する仕組みが取られることが多い。

また、ISO/IEC において、標準化も行われている [60, 61]。

■**リング署名** リング署名では、署名者があるグループの一員であることは明示的に分かるが、グループ内の誰であるかまでは分からない匿名性を保証する署名方式である。一般の署名方式では不可能であった、署名者のプライバシー保護、匿名性が必要となるアプリケーションにおいて需要がある。この意味では、グループ署名と同じ効果を有する署名方式である。ただし、グループ署名とは異なり、グループ管理者を置く必要がなく、署名生成時に柔軟にグループを選ぶことができるという点で、グループ署名とは異なっている。

グループ管理者を置かないことにより、グループ生成を柔軟にできることはメリットであるが、逆に、署名者の誰かが不正を行った場合に、不正を行った署名者が誰かを特定することは困難である、というデメリットもある。グループ署名を使用するか、リング署名を使用するかは、その利用環境に応じて決められるべきことである。例えば、署名を生成できるユーザを管理者が認可したユーザだけに絞りたい場合はグループ署名を使用する。この場合、管理者に対して、署名者の匿名性を担保することは困難となる。これに対し、リング署名の場合は完全な匿名性を担保したい場合に使用する。グループ署名とは逆に、署名者を管理する管理

者がいないため、誰でも署名を生成することができる。

リング署名は、実用事例は少ないが、活動が盛んになっている暗号資産において使われている [62]。

■しきい値署名 しきい値署名は、「シェア」と呼ばれるある情報の分散情報を利用する。まず、一つの平文に対し、複数の署名鍵のシェアを用いて署名のシェアを生成する。この署名のシェアをあるしきい値以上集めて、結合することで真の署名を生成する方式である。検証鍵は一つだけであり、各署名シェアではなく、真の署名の検証時に利用する。平文の正当性を一人の署名だけで確認するのではなく、しきい値以上の複数人の署名、すなわち、複数人が個々に平文を確認し、署名することにより、複数人が確認した平文となる。しきい値署名では、この複数人が生成した署名を、一度に検証することができる。

暗号資産における鍵漏洩への対策としてしきい値署名が使われることが多くなっている。例えば、しきい値署名の利用例として Bitcoin 等での利用が挙げられる。Bitcoin ではマルチ署名のような複数の署名者によるトランザクションの生成が可能である。しかし、この方法では複数の署名を必要とするため、ストレージ効率が悪いことや、署名者の構成が公開されてしまう等のデメリットがある。一方、しきい値署名はオフチェーンで利用することが前提となるが、複数の署名者がしきい値署名を使って署名の断片を作り、その署名を集めて一つの有効な署名としてトランザクションを生成する。この署名は単一の署名値でありマルチ署名に比べてストレージ効率が良く、署名者の構成が署名値からは分からない。ここでのオフチェーンとは直接ブロックチェーンの処理に組み込まれることの無い処理のことを指している。

そして、このオフチェーン処理の署名について、特に Bitcoin が利用する署名方式を ECDSA からシュノア署名に置き換わるにあたり、しきい値署名の利用が推進される可能性が種々報告されている [63]。また、暗号資産 quredo のイエローペーパーではしきい値署名が使用され [64]、フレセッツ社では HSM を利用したしきい値署名 [65] による Ethereum 秘密鍵保護方式 [66]、Coinbase 社では DApp Wallet [67] と呼ばれる暗号資産用のアプリで採用しているようであり、分散型のブロックチェーンを開発するプロジェクトである DFINITY ではスマートコントラクト canister が保持する Bitcoin や Ethereum の鍵をしきい値 ECDSA にする提案 (V.Shoup) [68] 等の報告がある。

これに応じて、標準化活動もおこなっている。アメリカ国立標準技術研究所 (National Institute of Standards and Technology, NIST) では、2019 年にしきい値暗号に関する NIST の標準化の目的等について記述された NISTIR 8214 [69] が

発行され、さらに 2020 年にそのロードマップとなる NISTIR 8214A が発行された [70]。そして、2021 年 9 月にフィードバックの締切があり、2022 年 8 月に開発者向けガイドラインと推奨の 8214B が発行された [71]。また、IETF では、標準化提案が行われ、標準化プロセスが始められている [72]。

2.4.3 その他の高機能暗号の活用事例と標準化動向

■**秘密分散** 秘密分散法は、その名の通り秘密情報を何らかのグループのメンバー間で分散する暗号技術の一種である。秘密鍵などの秘密情報を保管する際は、その秘密情報の紛失、破損、盗難、漏洩といった脅威を考慮する必要がある。破損や紛失については情報のコピーを作ることによって対策できるが、コピーが増えれば増えるほど盗難や漏洩のリスクが高まる。これを解決するため、秘密分散法は秘密情報を「シェア」と呼ばれる複数のデータに分割し、これを複数の当事者間で分散して保管する。このシェアは、ある定められた組み合わせを揃えたときのみ元の秘密情報が復元でき、また、復元の条件を満たさないシェアを揃えても、元の秘密に関する情報が漏れないように作られる。最もシンプルでよく用いられる「組み合わせ」の例はしきい値型である。すなわち、 n 個のシェアのうち、 k ($\leq n$) 個のシェアを集めることで秘密が復元できるような方式である。

上述の特徴の通り、秘密分散法の直接的な応用は分散ストレージサービスである。秘密分散法を用いて分散保管することで、データの秘匿性を保証しながら冗長性を確保することが可能となる。そして、この冗長性の確保により、一部のストレージに障害が起きた場合への耐性を取ることが可能となる。具体的なアプリケーションとしては、文書管理システムやコンテンツ配信システムなどへの適用事例がある [73]。自然災害の多い日本国においては、事業継続計画 (Business Continuity Plan, BCP) 対策は重要な課題であると認識されており、とりわけ東日本大震災の直後は、Azure 等のクラウドストレージサービスを利用しながら世界中のサーバにデータを秘密分散するサービス [74] が登場するなど、セキュリティと災害対策を両立する技術として活用されている。

また、計算量的な秘密分散法として知られる All-Or-Nothing Transform (AONT) を利用したものでは、Shin らによる AONT [75] を用いたサービスが ZenmuTech 社によって実装され、近年実用化されている [76]。

秘密分散法の応用は分散保管にとどまらず、様々な暗号プロトコルの要素技術としても用いられている。特に、複数のエンティティで一つの計算を行うマルチパーティ計算との関連が深い。近年ではブロックチェーンと組み合わせた秘密鍵の分散管理システムなどへの応用もあり、マルチパーティ計算と組み合わせるこ

とで暗号資産等の秘密鍵を安全に運用する手段として注目されている [77]。

標準化も進められており、ISO/IEC SC27 WG2 においては、秘密分散法の定義、記法、代表的な方式の標準規格が発行されている [78]。現在の規格では、5つの代表的な方式が記載されている。1つ目は、しきい値型の完全秘匿性を達成する Shamir の秘密分散法 [79] であり、今日においては最もよく知られた方式である。完全秘匿性とは、攻撃者が無限の計算能力を持っていたとしても破ることができない秘匿性のことである。2つ目は、完全秘匿性の要件を緩和する代わりに、シェアのサイズを秘密情報のサイズより小さくするランプ型秘密分散法 (Ramp Secret Sharing) [80, 81] であり、Shamir の秘密分散法の一般化として構成される。3つ目は、加法型秘密分散法 (Additive Secret Sharing) と呼ばれる手法に基づく、一般的なアクセス構造を実現する秘密分散法 [82] である。これは、秘密情報へのアクセスを許されるシェア集合と、許されないシェア集合を指定可能な秘密分散法であり、しきい値型よりも柔軟なアクセス構造を導入することが可能である。4つ目は、複製型秘密分散法 (Replicated Secret Sharing) と呼ばれる手法に基づく、一般的なアクセス構造を実現する秘密分散法 [83] である。この方式も、任意のアクセス構造を実現することが可能である。こちらの方式はシェアのサイズがパーティ数に対して大きいですが、方式の特殊ケースは近年の効率的なマルチパーティ計算にも利用されている [84, 85]。5つ目は、計算量的安全な秘密分散法 [86, 87] である。これは、情報理論的に安全な秘密分散法と共通鍵暗号を組み合わせることで実現でき、情報理論的な秘密分散法と比較してシェアサイズが大幅に削減できることが知られている。

■マルチパーティ計算 マルチパーティ計算 (Multi-Party Computation, MPC) は、複数の参加者 (マルチパーティ) が自身の秘密情報を秘匿しながら、それらの秘密情報を入力とする種々の関数をパーティ間で協調計算することを可能とする技術である。MPC には、大きく分けて秘密分散法をベースにしたものと、Garbled Circuit をベースにしたものがある。前者は後者に比べて参加者間で複数回の通信を必要とする代わりに、計算量・総通信量が小さい傾向にある。また、両者ともに参加者のうち一定数が結託しないという仮定の下で安全性が保証される。そして、標準的な公開鍵暗号では困難であった暗号化しながらの計算処理が、MPC では可能となる。また、準同型暗号と比較した場合、通信量や計算量に優れるため、処理のスループットが高いという特徴がある。

秘密分散法をベースにした MPC 方式の活用事例として、個人の秘密情報を秘匿したまま、統計情報の計算・機械学習等のデータ分析を行うサービス提供がある [88]。例えば、病院の臨床データの分析のために、ディープラーニングをする

際、入力に対してマルチパーティ計算させることで、平文の入力データを誰にも知られることなく、分析結果を得ることができる。秘密分散法がベースであるため、前述の秘密分散の項で説明したように、入力データはシェアと呼ばれるデータに分割され、複数の入力データシェアが作成される。その後、マルチパーティ計算を行う複数のサーバにそれぞれの入力データシェアが入力される。入力シェアを得たサーバは、ディープラーニング用にサーバ独自の計算を行うとともに、別のサーバと通信をすることで、分析結果データを分割したものとなる出力シェアを生成する。この出力シェアを全部集めて合成することで、分析結果を得る方法である。

さらに、複数の医療機関が保持するゲノム情報や診療情報を統合して解析することにも応用されている [89]。このゲノム情報は非常に機微性の高い情報であり、漏洩は絶対に許されない。そこで、MPC を用いてデータ秘匿性を守りながらデータを収集し、医療・医学に活用する方法としている。

さらに、国外でも、政府の持つ TAX Records と STUDENT Records より、教育と年収の関係、大学在学中のアルバイトと留年の関係、大学の学問分野とアルバイト量の関係を分析 [90] したり、エネルギー消費の最適化を行うために大手企業の産業データを秘匿化して収集し分析するエネルギー供給の最適化 [91] 等、多くの応用例がある。また、国内外で多くのスタートアップ等による活用が進んでおり、コミュニティの形成を通じた知見の共有、企業間の連携が進んでいる [92]。さらに、秘密計算技術が広く社会実装され、クラウドサービスのデータ保護に対する不安の払拭や、組織や企業の枠を超えたデータ利活用により新たな価値が創出されることを目的とし、様々な秘密計算方式を俯瞰した実用的かつ客観的な安全性基準や、ユーザが秘密計算を活用する際の参考となる指針の検討、ホームページやイベントを通じた、上記基準や指針の検討状況、方式の性能等に関する技術資料や先端事例の情報発信 [93]、秘密計算を中心としたプライバシー保護関連技術に関連する情報発信、および、個人情報保護法の改正を始めとしたデータ活用とプライバシー保護がコンフリクトしている現状に対応すべく、法令遵守したデータ活用やプライバシー保護テクノロジーの勉強会や情報発信を行っている団体もある [94]。さらに、NIST はプライバシーに関するプロジェクト Project on Privacy-Enhancing Cryptography (PEC) の中で、MPC の普及に務めている [95]。

標準化も進められており、現在、ISO/IEC SC27 WG2 において、秘密分散型 MPC の定義、記法、代表的な方式の標準化に向けたドラフト作成が進められている [96]。

Garbled Circuit をベースにした方式では、2 者計算の様々なシーンで利用されるが、近年では鍵管理の文脈において、秘密鍵を分散して保護しながら、それを

Garbled Circuit によって復元することなく暗号化・復号・署名等の計算を可能とする Hardware Secure Module (HSM) の代替としての応用が存在する。特に、暗号資産における電子署名の鍵管理手法として注目を集めており、従来の取引所等が集中的に鍵を管理する形態と比べて、鍵管理を集中して行う必要がなくなり、取引所等の単一障害点が排除されており、秘密情報を含む情報漏洩に対しても耐性があり、安全性が高い利点がある [97]。

Garbled Circuit ベースの MPC についての標準化の動向はないが、先に紹介した NIST の PEC プロジェクトにおいては、PEC 技術の一種として Garbled Circuit に関するワークショップ等での報告が行われている [95]。

■**ゼロ知識証明** ゼロ知識証明とは、ある人が他の人に特定の命題を証明したいときに、証明したいこと以外の知識を与えることなく証明する手法であり、以下に示す完全性、健全性、ゼロ知識性の3つの性質をもつ。

- 完全性：証明者の命題が真ならば、検証者は真であることが必ずわかること。
- 健全性：証明者の命題が偽ならば、検証者はかなり高い確率で、偽であることを見抜けること。
- ゼロ知識性：あらゆる場合で、検証者が証明者から何らかの知識を得ようとしても、証明者の命題が真であること以上の知識は得られないこと。

ここで証明したい命題には、ある値が規定の範囲内に存在するかの「範囲の証明」や、任意の演算が正しく実行されたか「演算の証明」がある。一番の特徴は、証明したいこと以外の知識を与えないため、安全性が高いことである。

この特徴を利用して、いろいろな場面での活用が考えられている。例えば、不動産会社と顧客と金融機関の間での取引において「不動産会社は、顧客に対して『住宅を借りるために十分な所得があるか』を証明してほしいと考えている。顧客は、正確な所得金額を明かしたくない。顧客は、命題『一定金額以上の所得がある』ことを、金融機関を通して、ゼロ知識証明により証明する。」

より具体的には、銀行において、顧客の本人確認を行う時に、正確な数字を明らかにすることなく、給与が特定の範囲内にあることを証明したり (ZKRP)、特定の地域の住人であることを具体的な住所を明かすことなく居住地区の範囲を証明することをゼロ知識証明により行う。そして、ZKRP をオープンソースソフトウェアとしてリリースしているケースもある [98]。

また、匿名性のある暗号資産の Zcash においては、zk-SNARK と呼ばれるゼロ知識証明を用いて、トランザクションを生成することにより、送信アドレスと受信

アドレス、送金額を秘匿することができる [99]。

さらに、パブリックブロックチェーンの一番の課題は、BtoC もしくは BtoB でやりとりされるトランザクションの匿名性の確保である。そこで、パブリックチェーンの Ethereum 上でトークンのプライベートトランザクション（匿名送金）を可能にするゼロ知識証明を用いたプロトコルが開発されている [100]。ただし、プライベートトランザクションはデータサイズが大きくなり、Ethereum のシステムを稼働させるための手数料にあたる、いわゆる“gas 代”の負担が高くなる傾向にあるが、複数の証明のバッチ化の工夫により、コスト負担を軽減させている。

また、標準化を目的とした団体があり、その中には NIST も参加している団体もある。標準化では、ゼロ知識証明を用いたデータプライバシー製品やアプリケーションのセキュリティ保証と相互運用性を向上させることを目的としている。そして、作成されたドキュメントでは、理論から実装までの情報がカバーされている [101]。

■Oblivious Random Access Machine (ORAM) サーバに暗号化データを格納し必要に応じて読み込み/書き込みを行う場合に、データは秘匿されているが、暗号化データへのアクセス頻度等のアクセスパターンの情報からサーバに統計的な部分情報が漏れてしまう。Oblivious Random Access Machine (ORAM) においては、サーバにアクセスパターンを隠しながら、暗号化データの読み込み/書き込みが可能となる。具体的には、ORAM 上で暗号化データに対して読み込み/書き込みを行うたびに、格納位置をシャッフルするとともに再暗号化することで、各アクセス間の関係を秘匿する。

ORAM を利用して、プライバシー保護を可能としたデータベースサービスが開発されている [102]。このサービスでは、プライバシー保護の安全性指標である差分プライバシーの概念に従ったプライバシー保護技術を利用して個別のデータを明らかにすることなくデータ分析を可能とし、ORAM によりアクセスパターンの漏えいを防ぐことができる。また、暗号資産のトランザクションにおいて、マイナーによりトランザクションの順番を、マイナーに有利な順番に操作する攻撃が考えられている。このような攻撃を防ぐために、ORAM を利用した仕組みが開発されている [103]。この攻撃では、マイナーはユーザのアクセスを監視して攻撃を行うが、ORAM を用いることによってアクセスパターンをマイナーに秘匿できることになる。

■Private Information Retrieval (PIR) ユーザがデータベース (DB) を検索し、ユーザに必要な情報を取得するサービスは多数ある。この DB 検索を行うためには、

DB に対してクエリと呼ばれる質問を入力し、その質問に対する回答として、DB からファイルやデータ等の検索結果を受け取っている。この際、ユーザは、“質問内容も個人情報であり、この内容を DB の管理者であっても知られたくない” という場合もある。このような情報秘匿を行うために創出されたプロトコルが Private Information Retrieval (PIR) である。そして、PIR は暗号方式そのものではなく、暗号方式を用いて、ユーザのプライバシーを保護した情報検索法である。

具体的には、ある DB が n 個のデータ $D = \{D_1, \dots, D_n\}$ を持っている時に、ユーザは質問 i にある処理を施しクエリ q_i を作成し、 q_i を DB に送る。この q_i からは i の情報は全く漏洩しないように工夫されている。そして、DB は i に関する情報を何も得られないが、 q_i を用いてある演算をすることで、検索結果を含む複数の候補を出力することができる。ユーザは、受け取った複数の候補から、所望の D_i を抽出する。全体として、DB が得られる情報から i については何の情報も得られないプロトコルとなっている。

具体的な提案としては、Wang らは安全で信頼できるハードウェアの存在を前提として、データベースのアウトソーシングに関する考察を行なった [104]。ここでは信頼できる第 3 者機関 (Trusted Third Party, TTP) の存在、非存在の両方について議論されている。当時の解決方法としては、AMD, HP, Intel, Microsoft 等が中心となっている Trusted Computing Group (TCG) の仕様に準拠したセキュリティチップ Trusted Platform Module (TPM) や Intel 社による LaGrande Technology などの採用を検討していたことが論文からは窺える一方で速度の面から効率的な実装は少なかったと判断される。

コンピュータシステム・ストレージ系の書籍 [105] において PIR を用いた活用事例でのセキュリティとプライバシーを考慮したデータマイニングが実利用される場合の懸念点が示されている。1) コントロール外のデータの扱い、2) 技術的に担保されていたとしても非匿名化の可能性、3) サービスに対する法域外の扱い、4) 法律と現在の技術の対応が未熟、という 4 点である。PIR を適用するデータはセンシティブな扱いを要することが容易に想像できるか、上記のような課題が存在していることに留意しなければならない。

近年では 2020 年に Mozaffari らによる PIR の改良方式 [106] が広く知られることとなった。この論文の中で以下のようなアプリケーションが列挙されており、幅広い適用が検討されていることが見て取れる。

- インターネット ドメインの登録
- Tor サーバのリレー情報の取得
- プライベート メディア配信

- プライバシーを保護する電子商取引アプリケーション
- オープン アクセス電子論文リポジトリでのプライベート クエリ
- メッセージング アプリケーション
- プライベート オンライン通知
- プライベート ファイル共有アプリケーション

また日本でもブロックチェーン技術への応用に関するプレスが 2020 年 6 月に発表されている [107]。Bitcoin 等で利用されている Unspent Transaction Output (UTXO) と呼ばれるトランザクション（価値の移動を表現するフォーマット）のデータベースが運用されていると仮定して、この DB に対して検索を行う際の検索クエリが流出するというプライバシー問題を解決するために PIR を利用するというアナウンスであった。ただし、Bitcoin 関連のイベントに合わせたプレスであることや、その後、このプレスを発行した組織は買収されるなどして、その後の動きなどではなく実際には実用化されていないと考えられる。

また、暗号資産のアクセスの秘匿化に使用する実装が報告され、実用化されている [107]。この実装では、Microsoft から公開されている PIR のオープンソースソフトウェア SealPIR [108] が使われている。

■検索可能暗号 一般的な共通鍵暗号や公開鍵暗号では、サーバに暗号化文書を格納した場合に、暗号化すると平文が秘匿されるため、ある暗号文に対応する平文があるキーワードを含んでいるか検索することができない。検索可能暗号においては、サーバにキーワードの暗号文と一緒に格納することにより、検索キーワードを含むかどうか以外の平文の情報をサーバに漏らさずに、暗号文のままキーワード検索が可能となる。

検索可能暗号がもっとも活用される分野として注目されているのはデータベースと言って良い。この場合、データベースには、暗号化されたデータと、暗号化されたキーワードが保存されている。利用者はデータベースに暗号化されたキーワードのクエリを送る。データベースは復号鍵を持っていないので、平文のクエリを知ることはできない。データベースでは、その暗号化されたクエリと保存されている暗号化キーワードのマッチングを行い、検索結果を得る。検索結果も暗号化されたデータであり、その平文データをデータベースに知られることはない。

暗号化されたクエリにより検索を可能とするデータベースは MIT による CryptDB [109] を代表に多くの事例がある。CryptDB は Paillier 暗号等、複数の暗号技術を用いて暗号化データベースを実現しており、Cipherbase や SEED、Microsoft 社のシステム等、複数の商用サービスに採用されている。この他にも、

学術研究をもとにした著名なオープンソースソフトウェアとして KafeDB [110] 等がある。国内の商用のソリューションとしては、日立製作所はクラウドサービスとして検索可能暗号技術を提供しており [111]、三菱電機もソフトウェアを開発している [112] 等、さまざまなアプローチで活用されている。また、Intel SGX 上で検索可能暗号を実現する技術 [113] 等、多くの実用化研究が行われている。

検索可能暗号の標準化については、IETF において過去に準同型暗号についてのプレゼンテーション等で検索可能暗号についで触れられていたことがあったものの、標準仕様として議論されるには至っていない。

2.4.4 活用事例からみた高機能暗号の利用方法

ここまで、高機能暗号技術がどのようなものであるか、という視点に立ち、その活用例、実施例、標準化活動を紹介した。本章では、どのようなサービス、応用方法があるか、という視点に立ち、そのサービス、応用にとって、どのような暗号方式で有用であり、実用化されたか、を紹介する。

■**プライバシー強化** 種々のサービスにおいてプライバシーを強化する目的で、秘密計算を利用した暗号を社会実装したケースがある。秘密計算とは、計算を行うサーバ等の装置に対し、ある加工されたデータを渡し、そのサーバ等の装置において計算の元となる値を知られることなく、その結果を計算する方法の総称である。代表的な方式として、準同型暗号（3.1.4 章参照）を用いた方式、秘密分散法（3.3.1 章参照）を用いた方式、マルチパーティ計算（Multi-Party Computation, MPC）（3.3.2 章、3.3.3 章参照）を用いた方式、装置内に組み込まれている信頼できる秘匿されたハードウェア（Trusted Execution Environment, TEE）を用いる方式等が広く知られている。表 2.2 に示す事例は、この秘密計算が有用であると考えられる事例である。これらの中から、数例を以下に紹介する。

例 1. ゲノム解析による実証実験

ゲノム解析に関し、秘密計算を利用した企業と大学の共同実験が行われている [114]。ゲノムは、個人の遺伝子情報の全体であり、究極のプライバシー情報と言える。これまで、ゲノム情報は個別の医療機関で所有しているが、医療機関を超えた連携は、プライバシー情報の漏洩につながるため容易ではなかった。より多くのデータを集めることで、正確でより広範なゲノム解析が可能となるが、このプライバシー保護のハードルを超える必要があった。そこで利用したのが、秘密計算である。

実験の概要を図 2.1 に示す。ここでは、各医療機関が持つデータを加工し、複数

表 2.2 秘密計算の応用事例

領域	説明	適用例
統計処理	典型的な統計計算（合計、平均、最大値、最小値）	売上集計、ログの分析、ゲノム分析
突合	複数のデータソースから同じキーを持つレコードを突合	マーケティング、ゲノム分析
スコアリング／ AI 推論	入力および AI モデル等を秘匿しながら計算	AML（不正送金）
マッチング	要求条件を秘匿しながら、条件の合うレコードを検索	秘密オークション、場外の証券取引
連合学習	連合学習時の AI モデルの保護	特に機微なデータを元にした AI モデルの保護
投票	投票者を秘匿しながら集計	選挙

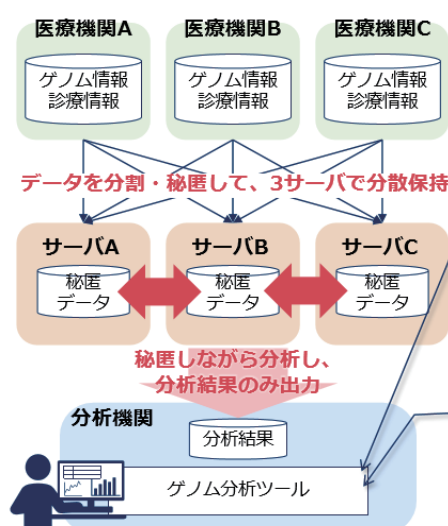


図 2.1 実証実験 1：ゲノム分析（[114] から引用）

の別のデータに変換した。具体的には、個々の医療機関が持つゲノム情報や、診療情報のデータを3分割し、秘匿化する。この分割されたデータから、元のゲノム情報や診療情報のデータは復元できないようになっている。いわゆる秘密分散法を用いた方式である。そして、これらの分割され秘匿化されたデータを、それぞれ異なる3台のサーバに分散して保存する。それぞれのサーバでは、受け取ったデータを利用するとともに他のサーバと通信しつつ、あらかじめ定められた演算を行う。これは、MPC技術の一種であり、入力データの加工方法である秘密分

散法と組み合わせ、秘密分散ベース MPC と呼ばれることもある。秘密分散ベース MPC では他のサーバとの通信を行うが、各サーバのデータが漏洩しないように、通信路の暗号化を追加する等、実運用時にはさななる安全性を担保する。

MPC では複数のサーバが必要となる。近年の個人情報漏洩のインシデントを見る限り、何らかのオペレーション時の個人の操作ミスによるものが多い。従って、複数のサーバの管理者を分けるだけでもインシデント対策として効果はある。さらに、サーバ毎に異なる組織、事業者などで管理することで、より安全なシステムが構築できる。

演算を終えた各サーバは、演算結果を分析機関に送る。分析機関では、すべてのサーバからの演算結果を統合する。この統合方法は、入力データの加工方法と、各サーバに定められた演算によって決まるが、秘密分散の逆演算とみなすことができる演算を行う。この秘密分散の逆演算を行うことにより、分析機関では、秘匿化されたデータの可読化を行う。

実験では、この可読化されたデータに対しゲノム分析ツールを用いて、典型的ゲノム分析、非典型的ゲノム分析の2種類の分析を行った。

- 典型的ゲノム分析（ゲノム変異分析）：ゲノム変異が起こる割合を年代や性別ごとに集計グラフ化する。
- 非典型的ゲノム分析（連鎖不平衡ブロック生成）：世代間の遺伝のし易さを変異場所のブロックで示す分析手法を利用。

これらの実験を通じ、秘密計算の処理の高速性と、分析に秘密計算が適用できることが実証された。

例2．創薬向けの実証実験

創薬向けの予測モデルの構築において、秘密計算を利用した方法の実証実験も行われている [115]。創薬においては、企業毎に種々の原料が使われ、その構造データは各企業の機密情報である。このため、データを企業から持ち出すことや、企業間で共有し、新たな創薬を行うことは困難であった。また、最近では、創薬にも機械学習が多く用いられている。機械学習では、学習によるモデルの構築が必要となるが、学習データが多いほど精度の高い学習が可能となる。このため、図 2.2 に示すような、秘密計算を用いた連合学習を行い、モデル構築の実証実験が行われた。

実証実験では、モデル構築に参加する複数の組織を想定している。各組織で個別のデータを用いて機械学習を行う。機械学習により出力されたモデルのパラメー

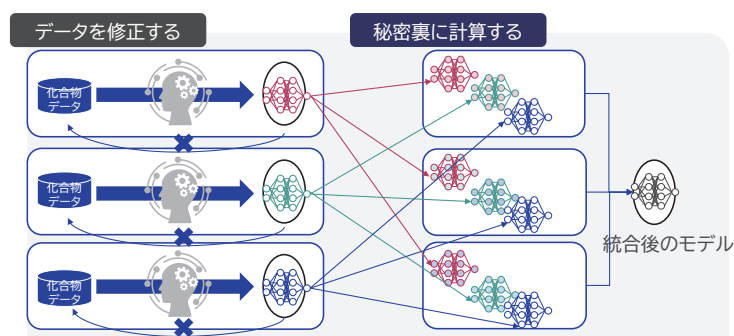


図 2.2 実証実験 2：創薬向け予測モデル（NEC 提供）

データを、秘密分散し、とりまとめを行う複数のサーバに送る。秘密分散を行う前のデータは、ある意味統計加工されたデータとなっており、ここから元データを知ることが困難である。複数のサーバでモデルを統合するが、その際に秘密計算を利用して統合モデルを作成する。サーバでモデルを統合する場合には、センターと各組織の間で通信するデータはモデルそのものではなく、差分値だけを扱うことで盗聴耐性を持たせることも可能である。

この実証実験では、データを暗号化したまま計算処理ができる秘密計算を用いた複数組織間のデータ統合の有効性を検証した。

例 1 において、サーバへのデータ入力、データの加工方法はデータを分割する方法が記されているが、加工方法は応用事例に応じて多種多様である。特に、個人のデータを利活用する場合、個人の同意を得ることが、個人情報保護法で求められている。ただし、個人のデータに対しある加工を施すことで、制限はあるものの個人のデータの利活用が可能となっている。この加工技術が、匿名加工や、仮名加工である。

- 匿名加工： 各医療機関が、複数のデータを一つにまとめ、外部への提出用データを作成する。データはまとめられ、加工データから、元データを知ることが不可能なため匿名加工したこととなる。
- 仮名加工： 各医療機関では、実名と匿名の対応表を作り、外部への提出用に匿名のデータ表を作成する。匿名化されており、実名を知ることが不可能なため、仮名加工したこととなる。

これらは、個人情報保護法に対応した個人データの利活用をするためにも、必須の技術となっている。

また、商用に供するために、各社で秘密計算のライブラリを整えている。MPC、準同型暗号を用いた方式、TEE を利用した方式など、利用者に合わせて利用できる。これらのライブラリを用いて、ゲノム解析、創薬の実証実験以外にも、

- 商店街の活性化のイベントを行う際に、その費用対効果をしるために、各店舗の売上情報を秘密計算を用いて統合分析する統計値処理
- 銀行における不正取引リスクを分析するために、各銀行が持つリスクデータを統合して分析するリスク分析の AI
- プラントのログ解析

等、多くの応用が検討されている。第 3 項目のプラントのログ解析は、個人情報保護とは直接は関係がない項目であるが、同じ要素技術で解決策を得られる項目例として、ここに挙げている。

■データへのアクセス制御－セキュアなパブリッククラウドの利用 企業、団体、組織（以後、組織）からの種々のサービス提供のみならず、組織内における電子データのやりとりにおいては、電子データは的確に保護されなければならない。これらの目的のために、属性ベース暗号や検索可能暗号と呼ばれる高機能暗号技術を利用して、データの保護、データへのアクセス制御を社会実装したケースがある。これらの事例を以下に紹介する。

例 1．組織内文書管理－属性ベース暗号技術の活用例

組織内の共有スペースに電子データを置き、データの共有をはかることはよく行われている。但し、電子データの中には、一部のメンバーだけで共有したいデータもある。この場合、共有するデータにアクセス制御をかける必要がある。さらに、暗号化により他のメンバーに閲覧できないようにすることで、強固なアクセス制御が可能となる。以下に紹介する社会実装例では、属性ベース暗号（3.1.2 章参照）を利用して、強固なアクセス制御を実現している。

属性ベース暗号とは、暗号文を復号するための条件を設定し、復号できるユーザを制御できる暗号技術である。復号鍵を所有しないユーザは、当然、暗号文を復号することはできない。さらに、復号鍵を持っているユーザであっても、条件に適合しない場合は復号できない。すなわち、暗号化とデータへのアクセス制御が一体化した技術である。

属性ベース暗号の論文での提案時より、活用例として、企業内や学校内での文書管理が記載されている。具体的には、企業内の役員だけが閲覧できる文書、部長以上の役職であれば閲覧できる文書、ある部署（例えば総務部のみ）の職員だけ

が閲覧できる文書、社員であれば誰でも閲覧できる文書など、文書の閲覧には閲覧できるユーザに条件がつけられていることが多い。ユーザの役職、所属部署などを属性と見て、職員 A さんは属性（人事部、課長）であり、所属 B さんは属性（総務部、部長）を持つとする。そして、職員 A さんには、属性（人事部、課長）に対応する復号鍵が渡され、人事部あての文書や、課長以上が閲覧できる文書、職員全員が閲覧できる文書が閲覧可能となる。職員 B さんは、属性（総務部、部長）に対応する復号鍵が渡され、総務部あての文書や、部長以上が閲覧できる文書、職員全員が閲覧できる文書が閲覧可能となる。文書の作成者は、閲覧者の属性を意識して、文書を暗号化することになる。ここで紹介した暗号技術が属性ベース暗号となる。より正確には、属性ベース暗号には大きく分けて 2 種類、暗号文に条件を入れる場合と、暗号鍵に条件を入れておく場合があり、ここで紹介した事例は暗号鍵に条件を入れた属性ベース暗号となる。

属性ベース暗号のその他の利用方法として、

- 複数の工場や事業所を所有する企業における、ロケーション／役職に応じた文書管理。
- クラウドサービスに文書を保管し、文書毎にアクセス可能な企業を制御できるファイル交換サービス。

などがあり、属性ベース暗号関連のソフトウェアがライブラリとして製品化されている [116, 117]。これらは、Windows ベースで開発されており、各種のインターフェース、ディレクトリサービスとの連携までも考えられている。

例 2. DB に保存されたデータの保護 – 検索可能暗号技術の活用例

組織内、組織外を問わず、電子データをサーバに置いて管理することはよく行われている。電子データはサーバ内のデータベース（DB）で管理され、ユーザからのデータ追加や検索などに応じている。通常、サーバもしくは DB の管理者は DB 内のデータを閲覧できる。悪意のある管理者であれば、データを悪用することは容易である。管理者の善意に頼るのみではなく、技術的な対策も求められる。さらに、近年はサイバー攻撃が多くなり、この対策も求められている。このため、DB のデータを管理者やサイバー攻撃者が知らない鍵によって暗号化することが対策として考えられてきた。さらに、ユーザからの検索キーワードも暗号化することで、何を検索しているのか等の情報も漏洩しないことが求められている。これらの要求に応える技術として、検索可能暗号（3.3.7 章参照）を利用した社会実装例を以下に紹介する。

検索可能暗号とは、データベース（DB）に暗号化されたデータを復号することなく検索できる暗号技術である。DB 内のデータはすべて暗号化されているため、DB が保管されているサーバが攻撃され、データが漏洩したとしても、安全性が担保できる。さらに、検索のためのキーワードも暗号化されているため、暗号化されていない通信路を経由しての検索であっても安全である。さらに、暗号化されたキーワードや、データをサーバ内で復号することがないため、サーバの管理者に対しても安全性が担保される。

従来、DB のデータは平文で管理されていた。もしくは、DB に平文データが入力された後、DB が暗号化を行い、暗号化されたデータを保存していた。前者のように DB で平文を管理する場合、検索も高速である。すなわち、平文の検索キーワードにより検索が行われるためである。ただし、DB にアクセス可能なユーザであったり、サイバー攻撃によって DB の中身が漏洩する場合を考えると、情報漏洩のリスクは高いものであった。また検索キーワードも平文であり、検索キーワードからも情報が漏洩していた。後者のように、DB が暗号化を行う場合は、DB に保存されているデータは安全に保存される。ただし、検索はやや遅くなる。すなわち、データの検索時には、保存されている暗号文を復号した後に、検索キーワードとのマッチングを行い検索結果を出力するため、暗号文を復号する時間が必要となる。さらに、セキュリティの観点からすると、DB の管理者が悪意を持つ場合、および、サイバー攻撃に DB がさらされた場合、情報漏洩のリスクが大きくなる。すなわち、どちらの場合も、DB が保存している秘密鍵の漏洩が考えられる。秘密鍵の漏洩は、DB が保存しているすべての暗号化データが平文データとなって漏洩することと同じ結果となる。また、メモリ解析もセキュリティリスクの一つであり、攻撃として考えられる。すなわち、検索処理は多くの場合、サーバのメモリ上で行われる。このため、管理者やマルウェアがメモリを監視することで平文情報がわかってしまう可能性がある。

これらのセキュリティリスクを回避する目的で開発された技術が検索可能暗号である。検索可能暗号を利用することで、DB に保存されているデータは暗号化され、さらに、復号鍵を DB は所持していない。すなわち、DB の管理者が悪意を持って情報を漏洩させた場合、および、サイバー攻撃に DB がさらされたとしても、漏洩するのは暗号化されたデータだけであり、復号鍵は漏洩しない。さらに、検索キーワードも暗号化されており、DB 内で復号されることはないため、検索キーワードからの情報漏洩もない。

検索可能暗号の実装例として、

- クラウドに DB 管理を委託した場合に、DB 内のデータを検索するセキュア

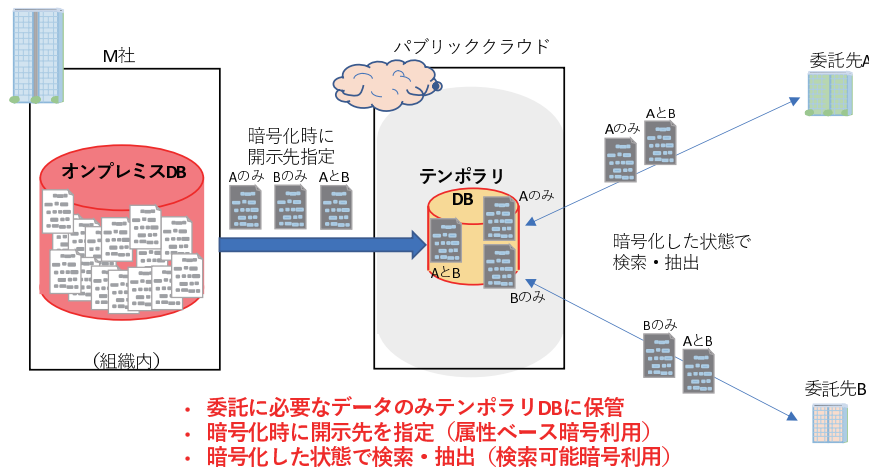


図 2.3 社会実装例：パブリッククラウドを用いたデータ共有

なアクセスを可能とする秘匿検索サービス

などがあり、検索可能暗号関連のソフトウェアがライブラリとして製品化されている [118]。

例 3. パブリッククラウドを用いたデータ共有

多くの企業、団体、組織（以後、組織）では協業の効率化等の目的でパブリッククラウドを利用する機会が多くなっている。例えば、自組織のサービスをパブリッククラウド上で展開したり、逆に、他組織のサービスを利用する際に利用することもある。さらに、パブリッククラウド上で展開されている機能／ライブラリ群なども多くなったこともあり、パブリッククラウドの活用が加速されている。このような状況下で、パブリッククラウドに置かれるデータの安全性を担保する目的で、属性ベース暗号と検索可能暗号の両方を社会実装したケースがあるので紹介する。

複数の組織で電子データを共有する場合にパブリッククラウドを利用することで、サービスの効率化を図れる場合がある。組織が所有する電子データが少ない場合、他組織と共有する電子データの量が少なく、関連する他組織の数も少なくなる。このような場合には、情報を共有するとしても、メールにデータを添付したり、記録媒体で渡すなどの、小規模な手段で情報共有が可能であった。これに対し、近年では、組織が持つ電子情報が膨大になり、他組織と共有する電子データの量が多くなり、関連する他組織の数も多くなり、状況が大きく変わっている。

例えば（図 2.3 参照）、ある組織（M 社）が業務効率化のために、他組織（A 社）と別他組織（B 社）に作業委託を行うことがあり、M 社と A 社、B 社は機密デー

タや顧客データを共有することもある。そして、A 社と共有する情報と B 社と共有するデータには、共通のデータもあり、異なるデータもある。そして、M 社は、社内にオンプレミス型の巨大な DB（オンプレ DB）が構築されていることはよくある状況である。委託先 A 社、B 社は M 社にとっては他組織であるため、M 社のオンプレ DB に A 社や B 社がアクセスすることは、委託に関係しないデータへのアクセスに対する危惧もあり、セキュリティ上の理由で好まれない。このため、パブリッククラウド（以後、サーバ）を利用して、オンプレ DB 中の共有すべきデータだけをサーバに置き、テンポラルな DB（テンポラリ DB）を作成し、A 社、B 社とデータ共有を行うことが考えられている。

テンポラリ DB を利用し、オンプレ DB 内の必要最小限のデータ共有であったとしても、M 社にとっては機密データであるため、このデータ共有を安全に行うことは重要である。さらに、利用するサーバの管理者は、容易にテンポラリ DB に置かれたデータにアクセスできる。さらには、サーバがパブリックであるが故に、サーバの管理者、M 社、A 社、B 社とは関係のない、不特定多数のユーザがアクセスする可能性もある。

このような状況であっても、テンポラリ DB に置かれた機密データを保護するために、属性ベース暗号と検索可能暗号を利用する。M 社が共有するデータをテンポラリ DB に置く際に、委託に必要なデータを抽出して、“A 社のみが閲覧できるデータ”、“B 社のみが閲覧できるデータ”、“A 社と B 社の両方が閲覧できるデータ”の 3 種類に分類する。そして、属性ベース暗号を利用して、“属性 A”、“属性 B”、“属性 A または属性 B”の属性を持つ復号鍵であれば、復号できるようにそれぞれのデータに対して暗号化を行い、暗号文をテンポラリ DB に置く。そして、A 社には“属性 A”を持つ復号鍵を、B 社には“属性 B”を持つ復号鍵を配布しておく。また、これらのデータを検索できるように、共有するデータからキーワードを抽出し、検索可能暗号を用いて、キーワードを暗号化し、テンポラリ DB に置く。A 社、B 社はテンポラリ DB のデータを閲覧する際、まず、検索キーワード（クエリ）を検索可能暗号を用いて暗号化して、暗号化クエリをテンポラリ DB に送る。テンポラリ DB では、検索可能暗号の検索機能を用いて検索結果を出力し、A 社もしくは B 社に返送する。

検索結果を受領した A 社もしくは B 社では、属性ベース暗号の復号鍵を用いて、受領したデータを復号するが、A 社が持つ復号鍵は“属性 A”を持ち、B 社が持つ復号鍵は“属性 B”を持つ。このため、A 社は、“A 社のみが閲覧できるデータ”と“A 社と B 社の両方が閲覧できるデータ”を復号することができるが、“B 社のみが閲覧できるデータ”を復号することはできない。逆に、B 社は、“B 社の

みが閲覧できるデータ”と“A社とB社の両方が閲覧できるデータ”を復号することができるが、“A社のみが閲覧できるデータ”を復号することはできない。

この一連の属性ベース暗号と検索可能暗号の実装により、データ閲覧に対するアクセス制御と、いわゆる秘匿検索、暗号化したままでの検索が可能となり、テンポラリDB内データに対する不要なデータ閲覧を防ぎ、キーワードからの情報漏洩を防いでいる。

参考文献

- [1] CRYPTREC: “電子政府における調達のために参照すべき暗号のリスト”. <https://www.cryptrec.go.jp/list/cryptrec-ls-0001-2012r6.pdf>.
- [2] S. Mitsunari: “MCL: a Portable and Fast Pairing-Based Cryptography Library”. GitHub Repository.
- [3] Stanford University: “The Pairing-Based Cryptography Library”. <https://crypto.stanford.edu/pbc/>.
- [4] Voltage: “Voltage SecureMail”. <https://www.microfocus.com/en-us/cyberres/data-privacy-protection/secure-mail>.
- [5] N. Suilvan: “Geo Key Manager: How It Works”. The Cloudflare Blog, <https://blog.cloudflare.com/geo-key-manager-how-it-works/>.
- [6] IETF: “RFC 5091: Identity-Based Cryptography Standard (IBCS)#1: Supersingular Curve Implementations of the BF and BB1 Cryptosystems”.
- [7] IETF: “RFC 5408: Identity-Based Encryption Architecture and Supporting Data Structures (January 2009)”.
- [8] IETF: “RFC 5409 : Using the Boneh-Franklin and Boneh-Boyen Identity-Based Encryption Algorithms with the Cryptographic Message Syntax (CMS)”.
- [9] IETF: “RFC 6267 : MIKEY-IBAKE: Identity-Based Authenticated Key Exchange (IBAKE) Mode of Key Distribution in Multimedia Internet KEYing (MIKEY)”.
- [10] IETF: “RFC 6539: IBAKE: Identity-Based Authenticated Key Exchange”.
- [11] IETF: “RFC 6507 : Elliptic Curve-Based Certificateless Signatures for Identity-Based Encryption (ECCSI)”.
- [12] IETF: “RFC 6508 : Sakai-Kasahara Key Encryption (SAKKE)”.
- [13] IEEE: “IEEE Std 1363.3-2013 - IEEE Standard for Identity-Based Cryptographic Techniques using Pairings”.

- [14] ISO/IEC: “ISO/IEC 18033-5:2015 - Identity-Based Ciphers”.
- [15] D. K. Smetters and G. Durfee: “Domain-Based Administration of Identity-Based Cryptosystems for Secure Email and IPSEC”, USENIX Security Symposium 2003 (2003).
- [16] S. Ruoti, N. Kim, B. Burgon, T. W. vander Horst and K. E. Seamons: “Confused Johnny: When Automatic Encryption Leads to Confusion and Mistakes”, Usable Privacy and Security, ACM (2013).
- [17] S. Ruoti, J. Andersen, T. Hendershot, D. Zappala and K. E. Seamons: “Private Webmail 2.0: Simple and Easy-to-Use Secure Email”, User Interface Software and Technology, ACM, pp. 461–472 (2016).
- [18] S. Meiklejohn, K. Mowery, S. Checkoway and H. Shacham: “The Phantom Tollbooth: Privacy-Preserving Electronic Toll Collection in the Presence of Driver Collusion”, USENIX security symposium, Vol. 201, No. 1 (2011).
- [19] L. Veltri, G. Morabito, S. Salsano, N. Blefari-Melazzi and A. Detti: “Supporting Information-Centric Functionality in Software Defined Networks”, IEEE ICC 2012 (2012).
- [20] M. Santos, B. D. Oliveira, C. Margi, B. Astuto, T. Turletti, I. S. Antipolis and K. Obraczka: “Software-Defined Networking Based Capacity Sharing in Hybrid Networks”, IEEE ICNP 2013 (2013).
- [21] M. Santos, B. Astuto, K. Obraczka, T. Turletti, B. D. Oliveira and C. Margi: “Decentralizing SDN’s Control Plane”, IEEE Conference on Local Computer Networks 2014 (2014).
- [22] 三菱電機 IT ソリューションズ: “パッケージプラス (R) トランスポーター”. https://www.mdsol.co.jp/products/general_affairs/transporter.html.
- [23] テキサス大学: “Ciphertext-Policy Attribute-Based Encryption : PBC (Pairing-Based Cryptography) ライブラリ (C ライブラリ) を利用したライブラリ”. <http://acsc.cs.utexas.edu/cpabe/>.
- [24] “Java Pairing-Based Cryptography Library (JPBC): [GGH+13] 方式を利用した Java による高機能暗号ライブラリ”. <http://gas.dia.unisa.it/projects/jpbc/>.
- [25] “OpenABE: [27] 方式の変形版を利用した C/C++ オープンソースライブラリ”. <https://github.com/zeutro/openabe>.
- [26] S. Garg, C. Gentry, S. Halevi, A. Sahai and B. Waters: “Attribute-Based

- Encryption for Circuits from Multilinear Maps”, CRYPTO 2013, Vol. 2, Springer-Verlag, pp. 479–499 (2013).
- [27] V. Goyal, O. Pandey, A. Sahai and B. Waters: “Attribute-Based Encryption for Fine-Grained Access Control of Encrypted Data”, ACM CCS 2006, pp. 89–98 (2006).
 - [28] AACS: “Advanced Access Content System (AACS) Introduction and Common Cryptographic Elements Book”. https://aacsla.com/wp-content/uploads/2019/02/AACS_Spec_Common_Final_0953.pdf.
 - [29] D. Naor, M. Naor and J. Lotspiech: “Revocation and Tracing Schemes for Stateless Receivers”, CRYPTO 2001, Springer-Verlag, pp. 41–62 (2001).
 - [30] <https://aacsla.com/>.
 - [31] NICT, 神戸大学, エルテス: “プライバシー保護深層学習技術を活用した不正送金検知の実証実験において金融機関 5 行との連携を開始”. <https://www.nict.go.jp/press/2020/05/19-1.html>, https://www.kobe-u.ac.jp/research_at_kobe/NEWS/collaborations/2020_05_19_01.html.
 - [32] “DeepProtect”. <https://deepprotect.nict.go.jp/>.
 - [33] L. T. Phong, Y. Aono, T. Hayashi, L. Wang and S. Moriai: “Privacy-Preserving Deep Learning via Additively Homomorphic Encryption”, **Vol.13, No.5**, pp. 1333–1345 (2018).
 - [34] 日本経済新聞: “Suica 履歴を「秘密計算」データの中身、読まずに分析”. <https://www.nikkei.com/article/DGXZQ0UC233NT0T20C21A6000000/>.
 - [35] P. Paillier: “Public-Key Cryptosystems Based on Composite Degree Residuosity Classes”, EUROCRYPT 1999, Springer-Verlag, pp. 223–238 (1999).
 - [36] Z. Brakerski, C. Gentry and V. Vikuntantathan: “(Leveled) Fully Homomorphic Encryption without Bootstrapping”, ITCS 2012, ACM, pp. 309–325 (2012).
 - [37] J. Cheon, A. Kim, M. Kim and Y. Song: “Homomorphic Encryption for Arithmetic of Approximate Numbers”, ASIACRYPT 2017, Vol. 1, Springer-Verlag, pp. 409–437 (2017).
 - [38] “HElib (Homomorphic Encryption library)”. <https://github.com/homenc/HElib>.
 - [39] Z. Brakerski: “Fully Homomorphic Encryption without Modulus Switching from Classical GapSVP”, CRYPTO 2012, Springer-Verlag, pp. 868–886 (2012).

- [40] J. Fan and F. Vercauteren: “Somewhat Practical Fully Homomorphic Encryption”. ePrint 2012/144.
- [41] “Simple Encrypted Arithmetic Library”. <https://www.microsoft.com/en-us/research/project/microsoft-seal/>.
- [42] “HEAAN”. <https://github.com/snucrypto/HEAAN>.
- [43] L. Ducas and D. Micciancio: “FHEW: Bootstrapping Homomorphic Encryption in Less Than a Second”, EUROCRYPT 2015, Springer-Verlag, pp. 617–640 (2015).
- [44] I. Chillotti, N. Gama, M. Georgieva and M. Izabachène: “Faster Fully Homomorphic Encryption: Bootstrapping in Less Than 0.1 Seconds”, ASIACRYPT 2016, Springer-Verlag, pp. 3–33 (2016).
- [45] “PLISADE”. <https://palisade-crypto.org/>.
- [46] “Lattigo”. <https://github.com/ldsec/lattigo>.
- [47] “CRYPTOLAB”. <https://www.cryptolab.co.kr/eng/product/heaan.php>.
- [48] “Homomorphic Encryption Standardization”. <https://homomorphicencryption.org/standard/>.
- [49] ISO/IEC: “ISO/IEC 18033-6:2019 - Homomorphic Encryption”. <https://www.iso.org/obp/ui/#iso:std:iso-iec:18033:-6:ed-1:v1:en>.
- [50] NuCypher: “Umbral: A Threshold Proxy Re-Encryption Scheme”. <https://raw.githubusercontent.com/nucypher/umbral-doc/master/umbral-doc.pdf>.
- [51] ITmedia: “東芝「デジタル貸金庫」”. <https://www.itmedia.co.jp/pcuser/articles/1211/21/news053.html>.
- [52] 川合, 小関, 柴田, 大松: “属性ベース署名を用いたブロックチェーン”, DICOMO2017, 情報処理学会, pp. 669–671 (2017). https://ipsj.ixsq.nii.ac.jp/ej/?action=repository_uri&item_id=190099&file_id=1&file_no=1.
- [53] 堀川, 高谷, S. Z. Fazekas, 山村: “属性ベース署名を用いた匿名シングルサインオンの提案” (2017). <https://www.kurims.kyoto-u.ac.jp/~kyodo/kokyuroku/contents/pdf/2051-26.pdf>.
- [54] S. Hirose and J. Shikata: “Non-Adaptive Group-Testing Aggregate MAC Scheme”, ISPEC 2018, Springer-Verlag, pp. 357–372 (2018).
- [55] S. Sato and J. Shikata: “Interactive Aggregate Message Authentication

- Scheme with Detecting Functionality”, AINA 2019, Springer Nature, pp. 1316–1328 (2019).
- [56] ITU-T: “Recommendation ITU-T X.1366: Aggregate message authentication schemes for Internet of things environment”. <https://www.itu.int/ITU-T/recommendations/rec.aspx?rec=14262&lang=en>.
 - [57] 日本ネットワーク・オペレーターズ・グループ: “アグリゲート署名を用いた bgpsec の改良”. <https://www.janog.gr.jp/meeting/janog43/program/bgpsec/>.
 - [58] ARMORY: “Armory Secure Wallet”. <https://www.bitcoinarmory.com/>.
 - [59] “ERC: Standard API for multisig wallet smart contracts #763”. <https://github.com/ethereum/EIPs/issues/763>.
 - [60] ISO/IEC: “ISO/IEC 20008-2:2013 - Information technology - Security techniques — Anonymous digital signatures — Part 2: Mechanisms using a group public key”. <https://www.iso.org/standard/56916.html>, <https://www.iso.org/obp/ui/#iso:std:iso-iec:20008:-2:ed-1:v2:en>.
 - [61] ISO/IEC: “ISO/IEC 11889 - ISO/IEC 11889-1:2015 - Information technology — Trusted platform module library — Part 1: Architecture”. <https://www.iso.org/standard/66510.html>.
 - [62] S. Noether: “Ring Confidential Transactions”. ePrint 2015/1098.
 - [63] P. Wuille, J. Nick and T. Ruffing: “Bitcoin Improvement Proposal 340” (2020). <https://github.com/bitcoin/bips/blob/master/bip-0340.mediawiki>.
 - [64] K. McCusker and B. Spector: “Qredo Network” (2020). <https://www.qredo.com/qredo-yellow-paper>.
 - [65] J. Doerner, Y. Kondi, E. Lee and A. Shelat: “Secure Two-party Threshold ECDSA from ECDSA Assumptions”. ePrint 2018/499.
 - [66] フレセッツ株式会社: “世界初、フレセッツが hsm を活用した完全オフライン環境でのセキュアなマルチシグ実装に成功” (2020). <https://prtimes.jp/main/html/rd/p/0000000017.000028896.html>.
 - [67] Coinbase 株式会社: “What is a Dapp Wallet”. <https://help.coinbase.com/en/coinbase/trading-and-funding/trade-on-dex/what-is-a-dapp-wallet>.
 - [68] DFINITY: “Threshold ECDSA Signatures”. <https://forum.dfinity>.

- org/t/threshold-ecdsa-signatures/6152.
- [69] NIST: “Threshold Schemes for Cryptographic Primitives: Challenges and Opportunities in Standardization and Validation of Threshold Cryptography”. <https://csrc.nist.gov/publications/detail/nistir/8214/final>.
 - [70] NIST: “NIST Roadmap Toward Criteria for Threshold Schemes for Cryptographic Primitives”. <https://nvlpubs.nist.gov/nistpubs/ir/2020/NIST.IR.8214A.pdf>.
 - [71] NIST: “NIST Roadmap Toward Criteria for Threshold Schemes for Cryptographic Primitives”. <https://csrc.nist.gov/news/2022/nist-requests-comments-on-ir-8214b-initial-public>.
 - [72] IETF: “Two-Round Threshold Signatures with FROST”. <https://datatracker.ietf.org/doc/draft-irtf-cfrg-frost/>.
 - [73] 保坂, 多田, 加藤: “秘密分散法とその応用”, 東芝レビュー, **62**, 7 (2007).
 - [74] “NRI: Windows Azure と秘密分散法を利用したセキュリティ性の高い「世界分散ストレージサービス」”, <https://cloud.watch.impress.co.jp/docs/column/cloud/493419.html>.
 - [75] S. Shin, S. Yamada, G. Hanaoka, Y. Ishida, A. Kunii, J. Oketani, S. Kunii and K. Tomomura: “An Extended CTRT for AES-256”, WISA 2019, Springer-Verlag, pp. 79–91 (2019).
 - [76] “ZENMU-AONT に関する論文が 国際会議で best paper premium award を受賞”, <https://www.atpress.ne.jp/news/191661>.
 - [77] “Alchemy - What is MPC wallet?”, <https://www.alchemy.com/overviews/mpc-wallet>.
 - [78] “ISO/IEC 19592-2:2017 Information technology - Security techniques - Secret sharing - Part 2: Fundamental mechanisms”.
 - [79] R. Cleve, D. Gottesman and H.-K. Lo: “How to Share a Quantum Secret”, Physical Review Letters, **83**, 3, pp. 648–651 (1999).
 - [80] G. R. Blakley and C. Meadows: “Security of ramp schemes”, CRYPTO 1984, Springer-Verlag, pp. 242–268 (1984).
 - [81] 山本: “ (k, L, n) しきい値秘密分散システム”, 電子通信学会論文誌, **J68-A**, 9, pp. 945–952 (1985).
 - [82] M. Itoh, A. Saito and T. Nishizeki: “Secret Sharing Scheme Realizing General Access Structure”, IEEE Globecom 1987, pp. 99–102 (1987).

- [83] R. Cramer, I. Damgård and Y. Ishai: “Share Conversion, Pseudorandom Secret-Sharing and Applications to Secure Computation”, TCC 2005, Springer-Verlag, pp. 342–362 (2005).
- [84] T. Araki, J. Furukawa, Y. Lindell, A. Nof and K. Ohara: “High-Throughput Semi-Honest Secure Three-Party Computation with an Honest Majority”, ACM CCS 2016, pp. 805–817 (2016).
- [85] K. Chida, D. Genkin, K. Hamada, D. Ikarashi, R. Kikuchi, Y. Lindell and A. Nof: “Fast Large-Scale Honest-Majority MPC for Malicious Adversaries”, CRYPTO 2018, Springer-Verlag, pp. 34–64 (2018).
- [86] R. Kikuchi, K. Chida, D. Ikarashi, W. Ogata, K. Hamada and K. Takahashi: “Secret Sharing with Share-Conversion: Achieving Small Share-Size and Extendibility to Multiparty Computation”, IEICE Trans., **E98-A(1)**, pp. 213–222 (2015).
- [87] H. Krawczyk: “Secret sharing made short”, CRYPTO 1993, Springer-Verlag, pp. 136–146 (1993).
- [88] 千葉大学病院, NTT Com: “秘密計算ディープラーニング」などの技術を活用した臨床データ分析”. <https://www.ntt.com/about-us/press-releases/news/article/2021/0208.html>.
- [89] NEC, 大阪大学: “医療データ活用”. <https://jpn.nec.com/rd/technologies/201805/index.html>.
- [90] CYBERNETICA: “収入と教育履歴の分析”. <https://sharemind.cyber.ee/big-data-analytics-protection/>.
- [91] Cosmian: “エネルギー供給の最適化”. <https://cosmian.com/use-case/>.
- [92] MPC Alliance. <https://www.mpcalliance.org/>.
- [93] 秘密計算コンソーシアム. <https://secure-computation.jp/>.
- [94] 秘密計算研究会. <https://securecomputing.jp/>.
- [95] NIST: “NIST Project on Privacy-Enhancing Cryptography (PEC)”. <https://csrc.nist.gov/projects/pec>.
- [96] ISO/IEC: “ISO/IEC 4922-1/2 Information security — Secure multiparty computation”. <https://www.iso.org/standard/80508.html>.
- [97] Curv. https://medium.com/@william_97682/curv-mpc-is-not-key-sharding-276b020130c4.
- [98] INGBank. <https://www.ingwb.com/en/insights/distributed-ledger-technology/>

- ing-launches-major-addition-to-blockchain-technology.
- [99] <https://z.cash/technology/>.
 - [100] Ernst & Young. https://www.ey.com/en_gl/news/2019/12/ey-releases-third-generation-zero-knowledge-proof-blockchain-technology-to-the-public-domain, https://assets.ey.com/content/dam/ey-sites/ey-com/ja_jp/topics/library/info-sensor/2020/12/pdf/ey-japan-info-sensor-2020-12-08.pdf.
 - [101] ZKProof Standards. <https://zkproof.org/>, <https://docs.zkproof.org/pages/reference/reference.pdf>.
 - [102] TechCrunch: “Kalepso Looks to Break into The Crowded Encrypted Database Space” (2018). <https://techcrunch.com/2018/11/29/kalepso-database-encryption-battlefield-berlin/>.
 - [103] Coinspeaker: “Decentralized Service Protocol, Automata Network, Launches a Solution for the Miner Extractable Value (MEV) Issue” (2021). <https://www.coinspeaker.com/automata-network-miner-extractable-value-conveyor/>.
 - [104] S. Wang, X. Ding, R. H. Deng and F. Bao: “Private Information Retrieval Using Trusted Hardware”, ESORICS 2006, Springer-Verlag, pp. 49–64 (2006).
 - [105] M. Melucci and R. Baeza-Yates: “Advanced Topics in Information Retrieval”, Vol. 33, Springer (2011).
 - [106] D. Augot, F. Levy-dit Vehel and A. Shikfa: “A Storage-Efficient and Robust Private Information Retrieval Scheme Allowing Few Servers”, CANS 2022, Springer International Publishing, pp. 222–239 (2022).
 - [107] “事業者向け暗号資産ウォレット開発のフレセツツ、暗号資産ウォレットユーザに対するプライバシー向上技術の開発に成功”. <https://prt看imes.jp/main/html/rd/p/000000014.000028896.html>.
 - [108] “Microsoft sealpir”. <https://github.com/microsoft/SealPIR>.
 - [109] MIT: “CryptDB”. <http://css.csail.mit.edu/cryptdb/>.
 - [110] “Kafedb: End-to-end structually-encrypted relational database system”. <https://zheguang.github.io/kafedb/>.
 - [111] 日立製作所: “秘匿情報管理サービス 匿名バンク”. <https://www.hitachi.co.jp/products/it/harmonious/cloud/service/tokumei/index.html>.

- [112] 三菱電機：“部分一致対応秘匿検索基盤ソフトウェア”. https://www.mitsubishielectric.co.jp/corporate/randd/list/info_tel/a29/index.html.
- [113] G. Amjad, S. Kamara and T. Moataz: “Forward and Backward Private Searchable Encryption with SGX”, European Workshop on Systems Security 2019, Vol. 4, pp. 1–6 (2019).
- [114] NEC, 大阪大学：“複数機関が保有するゲノム情報をプライバシー侵害リスクを抑えて解析できることを実証”. https://jpn.nec.com/press/201907/20190723_03.html.
- [115] NEC: “連合学習技術と秘密計算技術を用いた創薬における予測モデル構築に関する実証実験を実施”. https://jpn.nec.com/press/202203/20220311_01.html.
- [116] 三菱電機：“三菱情報セキュリティソフトウェア feileprotection”. <https://www.mitsubishielectric.co.jp/business/security/fileprotect/index.html>.
- [117] 三菱電機：“関数型暗号とクラウドサービスを利用した機密情報ファイル交換サービス「パッケージプラス (r) トランスポーター」の提供開始”. https://www.mdsol.co.jp/newsrelease/doc/information_20160125.pdf.
- [118] 三菱電機：“Mistyguard <検索可能暗号> cizouxlib を開発”. https://www.mdis.co.jp/news/press/2021_1028.html.

第 3 章

主な高機能暗号技術のアルゴリズム・ プロトコルとその性能

3.1 守秘を目的とした高機能暗号技術

3.1.1 ID ベース暗号

■特徴 ID ベース暗号 (Identity-based encryption, IBE) は、ID をはじめとする任意の文字列を公開鍵に利用可能とする公開鍵ベースの暗号化手法である。1984 年にそのコンセプトを提案した Shamir [1] により、公開鍵の情報として ID 情報を利用可能とすることから ID ベース暗号と名付けられたが、本質的には ID 情報の代表的な具体例である電子メールアドレスやその他の識別情報などが利用可能であり、さらには ID 情報のみならず任意の情報を公開鍵の情報として利用可能である。公開鍵である ID 情報から容易にその所有者を特定することができることから、(通常の公開鍵暗号では必要となる) 公開鍵暗号基盤 (Public-Key Infrastructure, PKI) が発行する公開鍵証明書が不要となる。一方で、公開鍵暗号とは異なり、IBE では暗号の作成者 (送信者) と秘密鍵をもつユーザ以外に、各ユーザの鍵生成を行う鍵生成センタ (Private-Key Generator, PKG) が存在する。このような機関を想定しなければ、誰でも任意の ID に対して秘密鍵を生成可能となり、安全性が担保できない。また、IBE を実現するうえで最も重要な安全性が結託耐性である。結託耐性とは、多数のユーザが結託し自身の ID の秘密鍵を共有したとしても、結託に関与していないユーザの ID 宛の暗号文の解読が難しいことを保証する安全性である。

さらに、IBE は他の暗号技術を構成するための暗号要素技術として利用価値が高い。例えば、IBE を要素技術として用いることで、秘密鍵の漏洩に耐性のある鍵隔離暗号 [2] や前方秘匿暗号 [3]、暗号文が復号可能となる期間を指定可能な時刻特定暗号 [4]、強い安全性を満たす公開鍵暗号 [5] 等、様々な暗号技術を実現できることが知られている。

■方式の歴史・進展 結託耐性を持つ IBE の効率的な実現は難しく、Shamir によって IBE のコンセプトが提唱された 1984 年から 15 年以上が経過して初めて、2000 年に境-大岸-笠原により ID ベースの鍵共有方式 [6] が日本国内で提案された。さらに、2001 年に Boneh-Franklin による IBE 方式 [7] が海外で提案された。これらの方式の特筆すべき点は、Pairing 写像が初めて暗号構成に利用されたことである。Pairing 写像は 2 入力 1 出力の写像であり、暗号用途では楕円曲線上の Pairing 写像を用いる。特に、Pairing 写像のもつその双線形性によって、IBE を始めとする多くの高機能暗号技術の実現が可能となった (双線形性については具体的な構成例で説明する) 実際、Pairing 写像の登場を皮切りに多くの (高機能) 暗号技術の

構成研究が発展することとなった。

IBE 研究においても、Pairing 写像を用いた構成手法が現実的な IBE の実現方法として主流であり、活用事例や標準化のほとんどに Pairing 写像を用いた構成手法が採用されている。Boneh-Franklin IBE の提案以降の主要な成果として、まず 2004 年の IBE 方式 [8] が挙げられる。この方式は Boneh-Franklin IBE とは異なり（現実には存在し得ない）理想的なハッシュ関数が構成に不要な一方で、保証できる安全性が弱く、現実的な状況（適応的な攻撃に対する安全性）を捉え切れていなかった。その後、2005 年に（理想的なハッシュ関数なしで）適応的な攻撃に対して安全な方式 [9]、2006 年には安全性の根拠となる計算量仮定は強いものの適応的な攻撃に対して安全かつ効率的なパラメータサイズを達成する方式 [10] と、立て続けに主要成果が提案された。Boneh-Franklin IBE に続くブレイクスルーとなった研究が、Waters [11] による Dual System Encryption と呼ばれる証明技法の開発である。安全性証明にのみ登場し実際には利用しない“不完全な”暗号文や秘密鍵を定義することで、標準的な計算量仮定の下で適応的な攻撃に対して安全かつ効率的なパラメータサイズを達成する方式を初めて提案した。このように、Pairing 写像ベース IBE 方式は効率的なパラメータサイズを達成し得るが、一方で Pairing 写像演算それ自体の計算負荷が高いという欠点がある。

また、Pairing 写像以外を用いる IBE の構成手法として有名なものに、格子構造を用いたものがある。代表的な格子ベース IBE 方式 [12, 13] は基本的に Pairing 写像ベース IBE 方式の構成アプローチを基にして作られており、格子構造のもつ耐量子計算機性が最大の魅力である。効率性の観点では、格子ベース IBE 方式はパラメータサイズが大きくなるものの、暗号化や復号が基本的に行列演算で実現可能という特長がある。格子以外の構成手法として、古典的な乗法巡回群を用いた Döttling-Garg による IBE 方式 [14] も重要な IBE 研究の一つである。Pairing 写像の登場まで IBE が実現されなかったことから、IBE は通常の公開鍵暗号で利用されるような基本的な代数的構造での実現は難しいと考えられていたが、この研究がその認識を覆すこととなり、IBE の実現可能性の意味で大きなブレイクスルーとなった。

学術研究としては更に高機能な ID ベース暗号が提案されており、PKG の秘密鍵生成の負担の分散を目的とした階層型 (Hierarchical) IBE [15] や、秘密鍵が部分的に漏洩したとしても安全性を保証可能な漏洩耐性 (Leakage-resilient) IBE [16]、効率的に秘密鍵の失効を実現可能な鍵失効機能付き (Revocable) IBE [17] 等、想定される機能は多岐にわたる。

■具体的な構成例 具体的な IBE 方式として、最も基本的かつ標準化もされている Boneh-Franklin IBE のアルゴリズムを紹介する。各ユーザは個別のユニーク ID (ID) を持つものとする。

Setup: 公開パラメータ MPK とマスター鍵 MSK を設定する。素数位数 p の乗法群を $\mathbb{G}_1, \mathbb{G}_T$ とし、 $\mathbb{G}_1$ の生成元を g とする。また、Pairing 写像を $e : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_T$ とする。 e は双線形性を満たす。すなわち、任意の $a, b \in \mathbb{Z}_p$ に対して $e(g^a, g^b) = e(g^b, g^a) = e(g, g)^{ab}$ を満たす。ランダムな元 $s \in \mathbb{Z}_p^*$ を選択し、 $g_{pub} = g^s$ を計算する。また、ハッシュ関数 $H_1 : \{0, 1\}^* \rightarrow \mathbb{G}_1$ と $H_2 : \mathbb{G}_T \rightarrow \{0, 1\}^n$ を定めておく。公開パラメータ MPK を $MPK = (p, \mathbb{G}_1, \mathbb{G}_T, e, n, g, g_{pub}, H_1, H_2)$ 、PKG のマスター秘密鍵 MSK を $MSK = s$ とする。

Extract(MPK, MSK, ID): ユーザ鍵 SK_{ID} を生成する。 $ID \in \{0, 1\}^*$ から $Q_{ID} = H_1(ID) \in \mathbb{G}_1$ を計算し、ユーザ鍵 $SK_{ID} = Q_{ID}^s$ とする。

Encrypt(MPK, ID, M): メッセージ M の暗号化を行い、暗号文 C を生成する。 $g_{ID} = e(Q_{ID}, g_{pub}) \in \mathbb{G}_T$ を計算し、ランダムな元 $r \in \mathbb{Z}_p^*$ を選択する。その後、メッセージ $M \in \{0, 1\}^n$ から暗号文 $C = (g^r, M \oplus H_2(g_{ID}^r))$ を作成する。

Decrypt(SK_{ID}, C): 暗号文 C を復号し、メッセージ M を復号する。暗号文 $C = (U, V)$ に対して、 $V \oplus H_2(e(SK_{ID}, U))$ を出力する。この時、正しい暗号文であれば $(U, V) = (g^r, M \oplus H_2(g_{ID}^r))$ であり、 $e(SK_{ID}, U) = e(Q_{ID}^s, g^r) = e(Q_{ID}, g^s)^r = g_{ID}^r$ であるから、 $V \oplus H_2(e(SK_{ID}, U)) = M$ が成り立つ。

3.1.2 属性ベース暗号

■特徴 属性ベース暗号 (Attribute-Based Encryption, ABE) は、暗号文または秘密鍵の一方に属性情報を、もう一方にアクセス構造を持たせ、復号時に属性情報に含まれる値がアクセス構造の条件を満たすかどうか判別することで、復号者の制御が実現可能な公開鍵暗号方式の一つである。一般に (狭義の) 公開鍵暗号や ID ベース暗号 (IBE) においては、暗号文とそれを復号可能な秘密鍵は 1 対 1 の関係を持つため、暗号文の生成時に復号できるエンティティは唯一に決定され、もし複数のエンティティに同一のメッセージを送付する場合には、エンティティ数に比例する暗号文の生成が必要である。一方で、属性ベース暗号ではアクセス構造を指定することで、単一の暗号文でメッセージの秘匿ときめ細かなアクセス

制御を同時に達成可能となる。表現可能なアクセス構造は用いる暗号方式により様々であるが、多くの方式では例えば AND, OR で記述された条件式を表現することができる。

属性ベース暗号の方式は、鍵ポリシ型 (Key-Policy, KP) と暗号文ポリシ型 (Ciphertext-Policy, CP) の 2 種類の型に大別される。鍵ポリシ型ではエンティティが持つ秘密鍵に、暗号文ポリシ型ではメッセージにそれぞれアクセス構造を埋め込み、もう一方に属性情報を埋め込む点異なる。そして、適応先のユースケースによってより効果的な型を選択する必要がある、例えば、ユーザや機器等のエンティティが固有に有する情報を元に制御したい場合には暗号文ポリシ型、暗号化対象のメッセージが固有に有する情報を元に制御したい場合には鍵ポリシ型の方式が適していることが多い。

一般に属性ベース暗号方式を構成する場合、セットアップおよび鍵発行を実行する主体である鍵生成局と、暗号化および復号を実行する主体であるエンティティが存在する。エンティティは、自身の有する属性情報（またはアクセス構造）を含む秘密鍵を鍵発行局から発行してもらい、自身のみが扱うように安全に保管する。暗号化については、公開情報を知ることができる任意のエンティティが実行することができ、メッセージにアクセス構造（または属性情報）を指定して暗号化する。復号については、エンティティが各自の秘密鍵と暗号文を用いて復号を行うが、属性情報とアクセス構造の関係が満足する場合には元のメッセージに復号できる。

■方式の歴史・進展 属性ベース暗号は、2005 年に ID ベース暗号の拡張である Fuzzy IBE が Sahai らにより提案され、そのコンセプトが提唱された [18]。Fuzzy IBE では、秘密鍵および暗号文にそれぞれ複数の ID が割り当てられており、各 ID 集合の間で定められたしきい値以上の ID が一致した場合にのみ復号可能である。この複数の ID を属性集合、一致する ID とそのしきい値をアクセス構造と見なすことで、現在の属性ベース暗号の一種と捉えられる。翌年 2006 年、Goyal らにより、より広いクラスの条件式を扱うことができるはじめての実用的な属性ベース暗号が提案された [19]。Goyal らの方式は鍵ポリシ型であり、秘密鍵に対して AND, OR, しきい値を用いた論理式で表現された条件式を設定し、また暗号文に属性集合を指定することで、暗号文に含まれる属性値が自身の持つ秘密鍵の条件式を満たすかにより、きめ細やかな復号者の制御が可能となった。さらに 2007 年には、Bethencourt らにより、暗号文に対してアクセス構造を設定することが可能な暗号文ポリシ型の方式がはじめて提案された [20]。

その後、安全性の研究においては、適応的攻撃に対しても安全な方式 [21] や標準モデルと呼ばれる現実的な仮定の下で安全な方式 [22] の研究等が進められ、ま

た量子計算機に対して安全な格子ベースの方式も提案されている [23]。条件式で使えるクラス拡張の研究においては、数値の大小比較 [24]、NOT 条件 [25] を利用可能な方式等、数多くの方式が提案されている。

さらに、前述した基本的な性質以外の機能性を追加した方式も提案されてきた。例えば、鍵発行処理をより柔軟にするものとして、ある秘密鍵で復号可能な暗号文のうち、その中の一部の暗号文のみを復号できる新たな秘密鍵を生成し、権限を移譲できる方式 [19] や鍵発行局を階層化した方式 [26]、複数の鍵発行局の秘密鍵を用いて復号制御ができるマルチオーソリティ方式 [27]、さらに特権的なルートを持たない分散型マルチオーソリティ方式 [28] 等がある。

■具体的な構成例: GPSW06 KP-ABE 本節では、AND, OR, および (k, n) -しきい値を条件式として記述でき、はじめてきめ細かなアクセス制御が可能となった KP-ABE である Goyal らによる GPSW06 の構成を示す [19]。

はじめに本方式のアクセス構造の表現方法であるアセスツリー $\mathcal{T}$ について説明する。アセスツリー $\mathcal{T}$ の葉ノード以外の全てのノードはしきい値ゲートを表現できる。ノード x に対して、子ノードの個数を num_x 、しきい値を $1 \leq k_x \leq num_x$ である値 k_x とする。 $k_x = 1$ の場合には OR ゲートを、 $k_x = num_x$ の場合には AND ゲートを、 $1 < k_x < num_x$ の場合には (k_x, num_x) -しきい値ゲートを表現できる。各ノード x の子ノードは順序付けされており、それぞれ 1 から num_x までのインデックスが割り振られている。また葉ノードでは属性値が設定されており、 $k_x = 1$ とする。

アセスツリー中のノードの取り扱いを容易にするため、いくつかのサブ関数を準備する。あるノード x に対して、 $parent(x)$ を x の親ノード、 $index(x)$ を親ノード $parent(x)$ から割り振られたインデックスを指す値とする。 $parent(x)$ の各子ノードはこのインデックスで一意に特定できる。また $att(x)$ を x の属性値で、 x が葉ノードのときにのみ値を取るものとする。

以降では、素数 p に対して、位数 p の乗法群を $\mathbb{G}_1, \mathbb{G}_T, \mathbb{G}_1$ の生成元を g とする。また Pairing 写像を $e : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_T$ とし、双線型性を満たす。また $\mathbb{G}_T$ の生成元を $g_T = e(g, g)$ とする。ラグランジュ補間において、 $\mathbb{Z}_p$ 上の元の集合 S で基底多項式を $\Delta_{i,S}(x) = \sum_{j \in S, j \neq i} (x - j)/(i - j)$ とする。

このとき、セットアップ (Setup)、暗号化 (Encrypt)、鍵生成 (Keygen)、復号 (Decrypt) の 4 つのアルゴリズムは以下の通りである。

Setup: 属性ユニバースを $\mathcal{U} = \{1, 2, \dots, n\}$ とする。それぞれの属性値 $i \in \mathcal{U}$ に対して、 $\mathbb{Z}_p$ 上から一様ランダムに選択した値を t_i とする。また値 $y \in \mathbb{Z}_p$ を一

様ランダムに選択する。このとき、公開パラメータ MPK およびマスタ秘密鍵 MSK は以下の通りとなる。

$$MPK := (T_1 := g^{t_1}, \dots, T_n := g^{t_n}, Y := g_T^y), MSK := (t_1, \dots, t_n, y)$$

Encrypt(M, γ, MPK): メッセージ空間を $\mathcal{M} := \mathbb{G}_T$ とする。また入力としてメッセージ $M \in \mathcal{M}$ 、指定する属性集合 $\gamma \subseteq \mathcal{U}$ とする。暗号文は、ランダムな値 $s \in \mathbb{Z}_p$ を選択し、

$$C_\gamma := (\gamma, C' := MY^s, \{C_i := T_i^s\}_{i \in \gamma})$$

となる。

Keygen($\mathcal{T}, MSK$): ある属性集合 γ に対して $\mathcal{T}(\gamma) = 1$ の場合に限り復号できるアクセスツリー $\mathcal{T}$ に紐づいた秘密鍵を出力する。これは $\mathcal{T}$ を表現するノード集合 $\mathcal{N}$ の全てのノードに対して、ルートノードから下位ノードに向かって順に次の計算をすることで得る。

各ノード $x \in \mathcal{N}$ において、しきい値 k_x から次数 $d_x := k_x - 1$ である多項式 q_x を定める。ルートノード r では、多項式 q_r は、 $q_r(0) = y$ とし、 q_r 上の他の d_r 個の値をランダムに選択する。非ルートノード x では、親ノード $\text{parent}(x)$ に割り振られたインデックス $\text{index}(x)$ から、多項式 q_x の定数を $q_x(0) = q_{\text{parent}(x)}(\text{index}(x))$ とし、また q_x 上の d_x 個の値をランダムに選択する。最後に、葉ノード x では、 $\text{att}(x) \in \mathcal{U}$ に対して、

$$SK_x := g^{q_x(0)/t_{\text{att}(x)}}$$

を計算し、これらがアクセスツリー $\mathcal{T}$ に対する秘密鍵、

$$SK_{\mathcal{T}} := \{SK_x \mid \text{att}(x), x \in \mathcal{N}\}$$

となる。

Decrypt($C_\gamma, SK_{\mathcal{T}}$): 暗号文 $C_\gamma := (\gamma, C' := MY^s, \{C_i := T_i^s\}_{i \in \gamma})$ 、秘密鍵 $SK_{\mathcal{T}} := \{SK_x \mid \text{att}(x), x \in \mathcal{N}\}$ を入力として、アクセスツリー $\mathcal{T}$ の各ノード x に対して、関数 $\text{DecryptNode}(C_\gamma, SK_{\mathcal{T}}, x)$ を再帰的に実行することで計算する。関数 DecryptNode は、計算結果として $\mathbb{G}_T$ 上の元または $\perp$ を出力する。

ノード x が葉ノードである場合、 $i = \text{att}(x)$ として、以下を得る。

$$F_x := \text{DecryptNode}(C_\gamma, SK_{\mathcal{T}}, x) = \begin{cases} e(SK_x, C_i) = g_T^{s \cdot q_x(0)} & \text{if } i \in \gamma \\ \perp & \text{if } i \notin \gamma \end{cases}$$

ノード x が非葉ノードである場合、ノード x の子ノード z_i の結果 F_{z_i} から関数 DecryptNode を計算する。まずはじめに、子ノードの内、 $F_{z_i} \neq \perp$ を満たす、ある k_x 個のノードを $S_x := \{z_{i_1}, \dots, z_{i_{k_x}} \mid i_j \in [1, \text{num}_x]\}$ とする。またこれらのノードのインデックスの集合 $S'_x := \{\text{index}(z_i) \mid z_i \in S\}$ とする。このとき、 S_x のノード数が k_x 個に満たない場合は、 $\perp$ を出力する。それ以外の場合は、

$$\begin{aligned} F_x &:= \text{DecryptNode}(C_\gamma, SK_T, x) \\ &= \prod_{z \in S_x} F_z^{\Delta_{\text{index}(z), S'_x(0)}} \\ &= \prod_{z \in S_x} (g_T^{s \cdot q_x(\text{index}(z))})^{\Delta_{\text{index}(z), S'_x(0)}} \\ &= g_T^{s \cdot q_x(0)} \end{aligned}$$

とし、多項式補間によりノード x における多項式 q_x の定数 $q_x(0)$ を得られる。

結果として、ルートノード r に対する $\text{DecryptNode}(C_\gamma, SK_T, r)$ は、再帰計算により、 $F_r = g_T^{sy} = Y^s$ が得られ、最後に、メッセージ $M' := C'/F_r$ が出力される。

3.1.3 放送型暗号

放送型暗号では、送信者は複数の受信者に暗号文を同時送信し、なおかつ、そのうちの任意の受信者集合に復号権限を指定できる。この仕組みは、従来の暗号方式を単純に組み合わせても実現可能であるが、その場合は受信者数に比例した暗号文長が必要となるため、非効率的となる。これに対し、放送型暗号では受信者数よりも漸近的に短い暗号文長で同等の機能を実現することが可能である。

放送型暗号には、誰でも公開情報のみを用いた暗号化を行うことが可能な公開鍵型と暗号化にも秘密情報を必要とする共通鍵型が存在する。公開鍵型については、楕円曲線上の Pairing 写像に基づき、暗号文サイズが受信者数によらず定数となる方式 [29] が設計されて以降、効率化についての理論的な検討が継続されており、2020 年には、Agrawal と山田により、暗号文と鍵サイズが現実的な仮定の下で定数サイズとなる初めての方式が提案されている [30]。一方、共通鍵型に関しては、木構造などの組み合わせ論的技法を用いてパラメータを圧縮する方法が主流である。公開鍵型の構成と異なり、重い計算負荷を要する代数的な構造は必要

としないため、ある程度高速な処理が可能であり、各利用者に求められる鍵情報のサイズは実用的な範囲に抑えられるものの、公開鍵型のように定数サイズにまで圧縮することは困難だと思われる。

■Boneh, Gentry, Waters による放送型暗号 本稿では具体的な方式の例として Boneh-Gentry-Waters 放送型暗号方式（以下、BGW 方式）[29] を記述する。BGW 方式は、結託耐性と受信者人数に劣線形（比例しない）短い定数長の暗号文長を達成した初めての方式であり、歴史的に非常に重要である。また、提案以来効率性や安全性の改善の検討が続いてはいるものの、当該方式は現時点においても最も効率の良い方式の一つである。なお、原論文では Key encapsulation mechanism (KEM) の形で方式が記述されているが、ここでは Encryption の形で方式を記述する。平文空間は G_T となっているが、一般の平文を暗号化するためには $e(g_1, g_n)^t$ を適当な鍵導出関数に入力して得られる文字列を共通鍵暗号の秘密鍵として平文を暗号化すればよい。鍵導出関数としては、例えば SHA のような暗号的ハッシュ関数や、Pair-wise independent ハッシュ関数を用いればよい。

以下、 G と G_T を素数位数 p をもつ群とし、 $e : G \times G \rightarrow G_T$ を付随する（対称）Pairing 写像とする。

Setup($1^\lambda, n$): セットアップアルゴリズムは、システムの人数 n とセキュリティパラメータ λ を入力とし、適切な Pairing 写像 e を選び、公開鍵を

$$PK = (g, g_1, \dots, g_n, g_{n+2}, \dots, g_{2n}, v = g^\gamma) \in G^{2n+1}$$

と設定する。ここで、 g は G のランダムな生成元、 $g_i = g^{\alpha^i}$ 、 α と v は Z_p のランダムな元である。また、 i 番目のユーザに対する秘密鍵を

$$d_i = g_i^\gamma$$

と設定する。最終的な出力は $(PK, d_1, \dots, d_n)$ である。

Enc(PK, S, m): 暗号化アルゴリズムは、ユーザ集合 S にむけて平文 $m \in G_T$ を暗号化するために、 PK, S, m の入力を受け、ランダムな $t \in Z_p$ を選択し、暗号文を

$$CT = (c_0 = g^t, c_1 = (v \prod_{j \in S} g_{n+1-j})^t, c_T = e(g_1, g_n)^t m) \in G^2 \times G_T$$

により計算し、出力する。

Dec(PK, S, i, d_i, CT): 平文を $m := c_T \cdot e(d_i, \prod_{j \in S, j \neq i} g_{n+1-j+i}, c_0) / e(g_i, c_1)$ により計算し、出力する。

BGW 方式は、暗号文長、秘密鍵長がシステム内のユーザ数に依存せず、定数長であるという利点があるが、公開鍵長はシステム内のユーザ数に比例して長くなるという欠点がある。[29] では、上記方式のほかに公開鍵長と暗号文長のトレードオフを実現する方式が記載されている。

■**安全性について** BGW 方式は選択的攻撃に対して結託耐性を持つことが、Pairing 群上の Bilinear Diffie-Hellman Exponent (BDHE) 問題と呼ばれる問題の困難性を仮定して証明可能である。結託耐性とは、システム上の復号権限を持たないユーザ全員が結託しても、暗号化された平文の情報を得ることが識別不可能性の意味でできないことを指しており、選択的攻撃とは攻撃者が公開鍵を得る前に攻撃目標である受信者集合を定める種類の攻撃を指している。BDHE 問題とは、対称 Pairing 群の元 $(g, h, g_1, \dots, g_\ell, g_{\ell+2}, \dots, g_{2\ell})$ が与えられた時に、 $e(g_{\ell+1}, h)$ とランダムな G_T の元の識別をするという問題である。ここで、 g, h はソース群の生成元で、 $g_i = g^{(\alpha^i)}$ であり、 α は Z_p 上でランダムにサンプルされる。また、システム内のユーザ数 n に応じてパラメータ ℓ は大きくなる。BDHE 問題に対しては、 ℓ が特定の条件を満たす際に Cheon 攻撃 [31] の適用が可能であり、特に n が大きい場合には、パラメータ選定における注意が必要である。

■**その他の放送型暗号方式について** 放送型暗号の概念は Fiat と Naor らにより定式化された [32]。共通鍵型の放送型暗号としては [33] が著名である。BGW 方式は選択的攻撃に対する安全性しか証明できていないことが欠点の一つであったが、より強い適応的攻撃に対する安全性を持つ方式がランダムオラクルモデルで Gentry と Waters によって [34]、標準モデルで Waters によって提案された [11]。Waters による方式は、decisional linear 仮定と呼ばれる BDHE 仮定よりも標準的な数論仮定に基づき安全性証明が与えられている。標準モデルでの方式はその後の研究によってさらに効率化されている [35]。

3.1.4 準同型暗号

準同型暗号は、データを暗号化したまま演算処理（四則演算等）ができる暗号技術であり、一般には公開鍵暗号の機能要件だけで、この性質が保証されるわけではない。準同型暗号は、実現できる演算内容により、単一演算型、Leveled 型、完全型に大別される。単一演算型は、加算または乗算のみ実現可能であり、単純な演算処理のみ実現可能である。例えば、加法準同型性をもつ暗号方式として Okamoto-Uchiyama 方式 [36] や、Paillier 方式 [37] が有名である。Leveled 型は加算と乗算をともに実現可能であるが、演算回路の深さに（特に乗算の深さに）上

限がある。なお、回路の深さとは入力ゲートから出力ゲートまでに最大の長さのことである。完全型は、加算と乗算を共に実現可能であり、かつ乗算の回数に制限がない。実際には、Leveled 型にブートストラッピングと呼ばれる技術を組み合わせることにより完全型が実現されることが多い。また、Leveled 型のうち、扱える計算回路の深さが浅いものは Somewhat 型と呼ばれることも多い。以下に、具体的な準同型暗号技術のアルゴリズムを紹介する。

■Boneh, Goh, Nissim による準同型暗号 具体的な準同型暗号方式の例として Boneh-Goh-Nissim 準同型暗号方式（以下、BGN 方式）[38] を記述する。BGN 方式は 2-DNF formula に対する準同型暗号方式として提案されているが、ここでは整数上の加算と一回の乗算が可能な準同型暗号として記述する。 G と G_T を合成位数 $n = q_1 q_2$ をもつ群とし、 $e : G \times G \rightarrow G_T$ を付随する（対称）Pairing とする。ここで、 q_1 および q_2 は素数である。また、 G は位数 q_1 の群 G_{q_1} と位数 q_2 の群 G_{q_2} の直積と同型であり、ここでは $G^{q_i} = \{g^{q_i} : g \in G\}$ を $G_{q_{3-i}}$ ($i \in \{1, 2\}$) と同一視することにする。

Setup(1^λ): セットアップアルゴリズムでは、適切な Pairing 写像 e を選択、および、 G のランダムな生成元 g と u を選択する。さらに、 $h = u^{q_2}$ を計算し、公開鍵を

$$PK = (n, g, h)$$

と設定する。また、秘密鍵は $SK = (q_1, q_2)$ である。

Enc(PK, m): 暗号化アルゴリズムは、(整数として符号化された) 平文 $m \in \mathbb{Z}$ を暗号化するために、 PK, m の入力を受け、ランダムな $r \in \{0, 1, \dots, n-1\}$ を選択し、暗号文を

$$CT = g^m h^r$$

により計算し、出力する。

Dec(PK, SK, CT): 復号アルゴリズムの入力 CT は、 G の元である場合と、 G_T の元である場合がある。前者は準同型的乗算を行う前の暗号文で、後者は準同型的乗算を行った後の場合である。どちらの場合も、復号アルゴリズムはまず CT^{q_1} を計算する。次に、 $\hat{g}$ を底とした CT^{q_1} の離散対数を計算し、 m を得る。ここで、 $CT \in G$ の場合は $\hat{g} = g^{q_1}$ であり、 $CT \in G_T$ の場合は $\hat{g} = e(g, g)^{q_1}$ である。復号が可能なためには m はセキュリティパラメータの多項式程度でなければならない。

次に、準同型演算を記述する。

$\text{Add}(PK, CT_1, CT_2)$: 二つの暗号文に対して加法を準同型的に行うためには、 CT_1 と CT_2 は同じ群 (G か G_T) の元である必要がある。両者ともに G の群である場合には、ランダムな $r \in \{0, 1, \dots, n-1\}$ を選択し $CT_1 \cdot CT_2 \cdot h^r$ を、両者ともに G_T の群である場合には、ランダムな $r \in \{0, 1, \dots, n-1\}$ を選択し $CT_1 \cdot CT_2 \cdot e(h, h)^r$ を計算する。

$\text{Mult}(PK, CT_1, CT_2)$: 二つの暗号文に対して加法を準同型的に行うためには、 CT_1 と CT_2 は G の元である必要がある。その場合にランダムな $r \in \{0, 1, \dots, n-1\}$ を選択し、 $e(CT_1, CT_2)e(g, h)^r$ を計算する。

■安全性について BGN 方式は、部分群判定問題の困難性を仮定すると IND-CPA 安全性をもつことが証明できる。部分群判定問題とは、 G のランダムな元 g 、 G_{q_1} のランダムな元 h が与えられた時、 G のランダムな元と G_{q_1} のランダムな元が識別不可能であるという仮定である。

■BGN 方式の後続研究について BGN 方式は、合成数位数 Pairing 群の演算が必要であるが、合成数位数 Pairing 群は素数位数 Pairing 群に比べ大きいパラメータが必要であり、計算効率が悪い。BGN 方式の改良として、素数位数 Pairing 群に基づく準同型暗号方式が Freeman によって提案され [39]、のちにさらなる改良方式が Attrapadung らにより提案された [40]。Attrapadung らは提案方式の実装も行っている。

■完全準同型暗号方式について 初の完全準同型暗号は Gentry によって 2009 年に提案された [41]。Gentry の提案は格子問題の困難性を利用していたものだったが、これが契機となり格子に基づく完全準同型暗号および Leveled 準同型暗号の研究が活発になった。代表的な方式としては、Brakerski らによる BGV 方式 [42]、Gentry らによる GSW 方式 [43]、Chillotti らによる TFHE 方式 [44]、Cheon らによる CKKS 方式 [45] などが挙げられる。CKKS 方式以外は、準同型演算の結果が復号によって正確に得られるタイプの完全準同型暗号方式だが、CKKS 方式は復号結果にエラーが加わるタイプの方式であり、復号結果にエラーが加わることを容認することで効率化を図っている。CKKS を含む復号にエラーが含まれるタイプの完全準同型暗号に関しては、セキュリティ上の懸念が指摘されており、利用シナリオやパラメータ選定等に注意が必要な場合がある [46]。

完全準同型暗号の著名な実装例としては、Homomorphic Encryption library (HElib) [47] や Simple Encrypted Arithmetic Library (SEAL) [48] が挙げられる。HElib は IBM 社がオープンソースで提供している準同型暗号のライブラリであり、BGV 方式と CKKS 方式が利用可能である。また、SEAL は Microsoft 社

がオープンソースで提供している準同型暗号のライブラリであり、Fan らによる FV 方式 [49] と CKKS 方式が利用可能である。

3.1.5 プロキシ再暗号化

プロキシ再暗号化は、あるユーザ宛の暗号文を、復号することなく別のユーザ宛の暗号文へ変換することが可能な暗号技術である。より具体的に、ユーザ A は、もう一方のユーザ B とプロキシ（代理人）と呼ばれる第三者を指定し、プロキシに対して「再暗号化鍵 $rk_{A \rightarrow B}$ 」を預託する。プロキシは、この再暗号化鍵 $rk_{A \rightarrow B}$ を使って、ユーザ A 宛の暗号文 ct_A の中身を知ることなく、ユーザ B 宛の暗号文 ct_B に変換することができる。

プロキシ再暗号化方式は、再暗号化鍵 $rk_{A \rightarrow B}$ の性質や暗号文が備える安全性レベルによっていくつかに分類可能である。以下、代表的な分類のいくつかを紹介する。

- 一方向/双方向変換: ユーザ A から B への再暗号化鍵 $rk_{A \rightarrow B}$ を用いて、ユーザ B 宛の暗号文をユーザ A 宛の暗号文に変換することができない場合は一方向、できる場合は双方向。
- CPA/CCA 安全性: 選択的平文攻撃に対してのみ安全であれば CPA 安全で、選択的暗号文攻撃に対しても安全であれば CPA 安全。
- シングル/マルチホップ: 再暗号化した暗号文をそれ以上再暗号化できなければシングルホップ、連続して再暗号化できればマルチホップ。

これらの性質については、用途に応じて選定を行う必要があるが、求められる性質によって処理性能に差が生じるため、慎重な判断が必要となる。なお、双方向変換可能で CPA 安全なマルチホップ方式が、最も高速な処理が可能となる。以下、Blaze ら [50] により初めて提案された双方向変換可能で CPA 安全なマルチホップな方式である BBS 方式を紹介する。BBS 方式は、ElGamal 暗号の簡単な変形で導かれ、その後のプロキシ再暗号化の構成の基盤となっている。

■Blaze, Bleumer, Strauss によるプロキシ再暗号化方式 [50] 以下、 $\mathbb{G}$ を素数位数 p を持つ群とし、 $g \in \mathbb{G}$ をその生成元とする。

KeyGen: 各ユーザは、鍵生成アルゴリズムを実行し、公開鍵 $pk = g^x \in \mathbb{G}$ と秘密鍵 $sk = x \in \mathbb{Z}_p \setminus \{0\}$ を生成する。

ReKeyGen(sk_A, sk_B): ユーザ A と B は、各々の秘密鍵 $sk_A = a$ と $sk_B = b$ を入力に再暗号化鍵生成アルゴリズムを実行し、再暗号化鍵 $rk_{A \rightarrow B} = b/a \in \mathbb{Z}_p$ を

プロキシに渡す。

$\text{Enc}(\text{pk}_A, \text{msg} \in \mathbb{G})$: 暗号化アルゴリズムを利用し、ユーザ A 宛の暗号文 $\text{ct}_A = (\text{msg} \cdot g^r, (\text{pk}_A)^r)$ を生成する。なお、 r はアルゴリズムがサンプルする乱数である。ここで、通常の ElGamal 暗号は $\text{ct}_A = (g^r, \text{msg} \cdot (\text{pk}_A)^r)$ であることに注意されたい。

$\text{ReEnc}(\text{rk}_{A \rightarrow B}, \text{ct}_A)$: プロキシは、再暗号化アルゴリズムを実行し、ユーザ A 宛の暗号文 $\text{ct}_A = (\text{ct}_1, \text{ct}_2)$ から、ユーザ B 宛の暗号文 $\text{ct}_B = (\text{ct}_1, (\text{ct}_2)^{\text{rk}_{A \rightarrow B}})$ を生成する。

$\text{Dec}(\text{sk}, \text{ct})$: 各ユーザは、復号アルゴリズムを実行し、まず自身宛の暗号文 $\text{ct} = (\text{ct}_1, \text{ct}_2)$ から $\text{msg}' = (\text{ct}_1)^{\text{sk}} / \text{ct}_2$ を計算し、平文 $\text{msg} = (\text{msg}')^{1/\text{sk}}$ を出力する。

暗号文 ct_A がユーザ A 宛であれば、 ct_A は $(\text{msg} \cdot g^r, g^{ar})$ であるため、秘密鍵 $\text{sk}_A = a$ を用いて、 msg が正しく復号できることが確認できる。また、 $\text{rk}_{A \rightarrow B} = b/a$ を利用することで、暗号文 ct_A の第 2 項目が、 g^{br} に変換できるため、プロキシがユーザ B 宛の暗号文 ct_B に正しく変換できていることも確認できる。 $\text{rk}_{B \rightarrow A} = (\text{rk}_{A \rightarrow B})^{-1}$ であることに注意すると、 $\text{rk}_{A \rightarrow B} = b/a$ が双方向変換に対応していることが確認できる。最後に、異なるユーザ C に対する再暗号化鍵 $\text{rk}_{B \rightarrow C}$ を渡されたプロキシは、 ct_B をさらにユーザ C 宛の暗号文 ct_C に変換できるため、マルチホップ性も満たす。

■BBS 方式の安全性について BBS 方式は、通常の公開鍵暗号が備える最も基本的な選択平文攻撃に対する安全性 (CPA 安全性) に類似した CPA 安全性を備える。通常の CPA 安全性では、攻撃者は公開鍵 pk_A 、および、自身が選んだ任意の平文 msg_0 と msg_1 のいずれかに関する暗号文 ct_A^* を入手し、 ct_A^* がどちらの平文を暗号化しているか識別できないことを要求する。プロキシ再暗号化方式においては、これに加え、任意の $n \in \mathbb{N}$ に対して、攻撃者は n 人の異なるユーザの公開鍵 $\{\text{pk}_{B_i}\}_{i \in [n]}$ と再暗号化鍵 $\{\text{rk}_{A \rightarrow B_i}\}_{i \in [n]}$ を与えられた上での、識別不能性を要求する。具体的に BBS 方式では、 $(g, g^a, \{g^{b_i}, b_i/a\}_{i \in [n]}, g^r, \underline{g^{ar}})$ の組みと、 $(g, g^a, \{g^{b_i}, b_i/a\}_{i \in [n]}, g^r, \underline{g^{r'}})$ の組みが識別不可能であることを意味する。ここで、後者の下線部に現れる r' は他の変数に依存しないランダムな $\mathbb{Z}_p$ 上の要素である。この組みを識別する問題は、Decisional Diffie-Hellman (DDH) 問題の困難性に帰着でき、BBS 方式は DDH 問題の困難性仮定に基づいて CPA 安全であることが示される。

■その他のプロキシ再暗号化方式 BBS 方式は、もし攻撃者が (ct_A^* 以外の) 暗号文を復号できる復号オラクルと対話できる場合、平文 msg_0 か msg_1 のどちらが暗号化されているかを自明に識別することができる。従って、BBS 方式は CPA 安全であるが、より強い選択暗号文攻撃に対する安全性 (CCA 安全性) を持たない。Canetti と Hohenberger [51] は、CCA 安全性を備える、双方向変換可能でマルチホップな再暗号化方式を提案した。BBS 方式と比べ、構成に双線型写像を要するため、安全性と引き換えに効率性が少し劣る。

双方向変換可能なプロキシ再暗号化方式は、再暗号化鍵 $rk_{A \rightarrow B}$ を生成する際にユーザ A と B の秘密鍵 sk_A と sk_B を必要とする。しかし、一方向変換可能なプロキシ再暗号化方式は、再暗号化鍵 $rk_{A \rightarrow B}$ を生成する際に、ユーザ A の秘密鍵 sk_A とユーザ B の公開鍵 pk_B だけが必要になり、 $rk_{A \rightarrow B}$ からユーザ B の秘密鍵 sk_B の情報が漏洩することはない。従って、ユーザ B の暗号文をユーザ A の暗号文に変更する必要があるアプリケーションにおいては、一方向変換可能なプロキシ再暗号化方式が双方向変換可能な方式と比べ、安全性を担保できるため適切である。そして、Libert と Vergnaud [52] は、CCA 安全で一方向変換可能なプロキシ再暗号化方式を双線型写像から構成し、花岡ら [53] は、特定の性質を満たすしきい値暗号に基づく一般的な構成を示した。いずれの方式もシングルホップであり、CCA 安全で一方向変換可能、かつ、マルチホップな効率的なプロキシ再暗号化方式は現段階では開発されていない。

参考文献

- [1] A. Shamir: “Identity-Based Cryptosystems and Signature Schemes”, CRYPTO 1984, Springer-Verlag, pp. 47–53 (1984).
- [2] Y. Dodis, J. Katz, S. Xu and M. Yung: “Key-insulated public key cryptosystems”, EUROCRYPT 2002, Springer-Verlag, pp. 65–82 (2002).
- [3] R. Canetti, S. Halevi and J. Katz: “A forward-secure public-key encryption scheme”, EUROCRYPT 2003, Springer-Verlag, pp. 255–271 (2003).
- [4] K. Paterson and E. Quaglia: “Time-specific encryption”, SCN 2010, Springer-Verlag, pp. 1–16 (2010).
- [5] D. Boneh, R. Canetti, S. Halevi and J. Katz: “Chosen ciphertext security from identity based encryption”, SIAM J. Computing, **36**, 5, pp. 1301–1328 (2007).
- [6] R. Sakai, K. Ohgishi and M. Kasahara: “Cryptosystems Based on Pairings”,

- SCIS 2000 (2000).
- [7] D. Boneh and M. Franklin: “Identity-Based Encryption from the Weil Pairing”, CRYPTO 2001, Springer-Verlag, pp. 213–229 (2001).
 - [8] D. Boneh and X. Boyen: “Efficient Selective-ID Secure Identity Based Encryption without Random Oracles”, EUROCRYPT 2004, Springer-Verlag, pp. 223–238 (2004).
 - [9] B. Waters: “Efficient Identity Based Encryption without Random Oracles”, EUROCRYPT 2005, Springer-Verlag, pp. 114–127 (2005).
 - [10] C. Gentry: “Practical identity-based encryption without random oracles”, EUROCRYPT 2006, Springer-Verlag, pp. 445–464 (2006).
 - [11] B. Waters: “Dual System Encryption: Realizing Fully Secure IBE and HIBE under Simple Assumptions”, CRYPTO 2009, Springer-Verlag, pp. 619–636 (2009).
 - [12] C. Gentry, C. Peikert and V. Vaikuntanathan: “Trapdoors for Hard Lattices and New Cryptographic Constructions”, STOC 2008, ACM, pp. 197–206 (2008).
 - [13] S. Agrawal, D. Boneh and X. Boyen: “Efficient lattice (H)IBE in the standard model”, EUROCRYPT 2010, Springer-Verlag, pp. 553–572 (2010).
 - [14] N. Döttling and S. Garg: “Identity-based encryption from the Diffie-Hellman assumption”, CRYPTO 2017, Springer-Verlag, pp. 537–569 (2017).
 - [15] D. Boneh, X. Boyen and E.-J. Goh: “Hierarchical Identity Based Encryption with Constant Size Ciphertext”, EUROCRYPT 2005, Springer-Verlag, pp. 440–456 (2005).
 - [16] S. S. Chow, Y. Dodis, Y. Rouselakis and B. Waters: “Practical leakage-resilient identity-based encryption from simple assumptions”, ACM CCS 2010, pp. 152–161 (2010).
 - [17] A. Boldyreva, V. Goyal and V. Kumar: “Identity-based encryption with efficient revocation”, ACM CCS 2008, pp. 417–426 (2008).
 - [18] A. Sahai and B. Waters: “Fuzzy Identity-Based Encryption”, EUROCRYPT 2005, Springer-Verlag, pp. 457–473 (2005).
 - [19] V. Goyal, O. Pandey, A. Sahai and B. Waters: “Attribute-Based Encryption for Fine-Grained Access Control of Encrypted Data”, ACM CCS 2006, pp. 89–98 (2006).
 - [20] J. Bethencourt, A. Sahai and B. Waters: “Ciphertext-Policy Attribute-

Based Encryption”, IEEE S&P 2007 (2007).

- [21] A. Lewko, T. Okamoto, A. Sahai, K. Takashima and B. Waters: “Fully Secure Functional Encryption: Attribute-Based Encryption and (Hierarchical) Inner Product Encryption”, EUROCRYPT 2010, Springer-Verlag, pp. 62–91 (2010).
- [22] B. Waters: “Ciphertext-Policy Attribute-Based Encryption: An Expressive, Efficient, and Provably Secure Realization”, PKC 2011, Springer-Verlag, pp. 53–70 (2011).
- [23] S. Gorbunov, V. Vaikuntanathan and H. Wee: “Attribute-Based Encryption for Circuits”, STOC 2013, ACM, pp. 545–554 (2013).
- [24] D. Boneh and B. Waters: “Conjunctive, Subset, and Range Queries on Encrypted Data”, TCC 2007, Springer-Verlag, pp. 535–554 (2007).
- [25] R. Ostrovsky, A. Sahai and B. Waters: “Attribute-Based Encryption with Non-Monotonic Access Structures”, ACM CCS 2007, pp. 195–203 (2007).
- [26] G. Wang, Q. Liu, J. Wu and M. Guo: “Hierarchical Attribute-Based Encryption and Scalable User Revocation for Sharing Data in Cloud Servers”, Computers and Security, Vol. 30, Issue 5, ACM, pp. 320–331 (2011).
- [27] M. Chase: “Multi-Authority Attribute Based Encryption”, TCC 2007, Springer-Verlag, pp. 515–534 (2007).
- [28] A. Lewko and B. Waters: “Decentralizing Attribute-Based Encryption”, EUROCRYPT 2011, Springer-Verlag, pp. 568–588 (2011).
- [29] D. Boneh, C. Gentry and B. Waters: “Collusion Resistant Broadcast Encryption with Short Ciphertexts and Private Keys”, CRYPTO 2005, Springer-Verlag, pp. 258–275 (2005).
- [30] S. Agrawal and S. Yamada: “Optimal Broadcast Encryption from Pairings and LWE”, EUROCRYPT 2020, Springer-Verlag, pp. 13–43 (2020).
- [31] J. Cheon: “Security Analysis of the Strong Diffie-Hellman Problem”, EUROCRYPT 2006, Springer-Verlag, pp. 1–11 (2006).
- [32] A. Fiat and M. Naor: “Broadcast Encryption”, CRYPTO 1993, Springer-Verlag, pp. 480–491 (1993).
- [33] D. Naor, M. Naor and J. Lotspiech: “Revocation and Tracing Schemes for Stateless Receivers”, CRYPTO 2001, Springer-Verlag, pp. 41–62 (2001).
- [34] C. Gentry and B. Waters: “Adaptive Security in Broadcast Encryption Systems (with Short Ciphertexts)”, EUROCRYPT 2009, Springer-Verlag,

- pp. 171–188 (2009).
- [35] J. Chen, R. Gay and H. Wee: “Improved Dual System ABE in Prime-Order Groups via Predicate Encodings”, EUROCRYPT 2015, Springer-Verlag, pp. 595–624 (2015).
 - [36] T. Okamoto and S. Uchiyama: “A New Public-Key Cryptosystem as Secure as Factoring”, EUROCRYPT 1998, Springer-Verlag, pp. 308–318 (1998).
 - [37] P. Paillier: “Public-Key Cryptosystems Based on Composite Degree Residuosity Classes”, EUROCRYPT 1999, Springer-Verlag, pp. 223–238 (1999).
 - [38] D. Boneh, E. Goh and K. Nissim: “Evaluating 2-DNF Formulas on Ciphertexts”, TCC 2005, Springer-Verlag, pp. 325–341 (2005).
 - [39] D. Freeman: “Converting Pairing-Based Cryptosystems from Composite-Order Groups to Prime-Order Groups”, EUROCRYPT 2010, Springer-Verlag, pp. 44–61 (2010).
 - [40] N. Attrapadung, G. Hanaoka, S. Mitsunari, Y. Sakai, K. Shimizu and T. Teruya: “Efficient Two-level Homomorphic Encryption in Prime-order Bilinear Groups and A Fast Implementation in WebAssembly”, AsiaCCS 2018, ACM, pp. 685–697 (2018).
 - [41] C. Gentry: “Fully Homomorphic Encryption Using Ideal Lattices”, STOC 2009, ACM, pp. 169–178 (2009).
 - [42] Z. Brakerski, C. Gentry and V. Vikuntantathan: “(Leveled) Fully Homomorphic Encryption without Bootstrapping”, ITCS 2012, ACM, pp. 309–325 (2012).
 - [43] C. Gentry, A. Sahai and B. Waters: “Homomorphic Encryption from Learning with Errors: Conceptually-Simpler, Asymptotically-Faster, Attribute-Based”, CRYPTO 2013, Springer-Verlag, pp. 75–92 (2013).
 - [44] I. Chillotti, N. Gama, M. Georgieva and M. Izabachène: “Faster Fully Homomorphic Encryption: Bootstrapping in Less Than 0.1 Seconds”, ASIACRYPT 2016, Springer-Verlag, pp. 3–33 (2016).
 - [45] J. Cheon, A. Kim, M. Kim and Y. Song: “Homomorphic Encryption for Arithmetic of Approximate Numbers”, ASIACRYPT 2017, Springer-Verlag, pp. 409–437 (2017).
 - [46] B. Li and D. Micciancio: “On the Security of Homomorphic Encryption on Approximate Numbers”, EUROCRYPT 2021, Springer-Verlag, pp. 648–677 (2021).

- [47] “HElib (Homomorphic Encryption library)”. <https://github.com/homenc/HElib>.
- [48] “Simple Encrypted Arithmetic Library”. <https://www.microsoft.com/en-us/research/project/microsoft-seal/>.
- [49] J. Fan and F. Vercauteren: “Somewhat Practical Fully Homomorphic Encryption”. ePrint 2012/144.
- [50] M. Blaze, G. Bleumer and M. Strauss: “Divertible Protocols and Atomic Proxy Cryptography”, EUROCRYPT 1998, Springer-Verlag, pp. 127–144 (1998).
- [51] R. Canetti and S. Hohenberger: “Chosen-Ciphertext Secure Proxy Re-Encryption”, ACM CCS 2007, pp. 185–194 (2007).
- [52] B. Libert and D. Vergnaud: “Unidirectional Chosen-Ciphertext Secure Proxy Re-Encryption”, PKC 2008, Springer-Verlag, pp. 360–379 (2008).
- [53] G. Hanaoka, Y. Kawai, N. Kunihiro, T. Matsuda, J. Weng, R. Zhang and Y. Zhao: “Generic Construction of Chosen Ciphertext Secure Proxy Re-Encryption”, CT-RSA 2012, Springer-Verlag, pp. 349–364 (2012).

3.2 認証・署名を目的とした高機能暗号技術

3.2.1 属性ベース署名

属性ベース署名は、署名者が持つ署名鍵が、署名を付与する平文に付随する条件を満たしている時に限り署名を生成でき、かつ、署名からは条件が満たされたという情報以外は漏れない署名方式である。より具体的に、属性ベース署名は、平文ポリシー型と鍵ポリシー型に大別されるが、前者は平文に条件、いわゆるアクセス構造が付随し、署名生成鍵に属性が付随する。平文と署名鍵に付随する情報を逆転させると後者になる。この際、属性（例えば、会社員、30歳以上、東京都在住）がアクセス構造（例えば、18歳以上の都民であるか？）を満たす時に限り、署名を生成することができる。例えば平文ポリシー型であれば、アクセス構造を満たす属性を所持する署名者が署名したという情報以外が漏れないため、匿名性を担保しながらアクセス制御が可能になる。どちらのポリシー型が適切かはアプリケーションによるが、一般的に平文ポリシー型の方が利便性が高い属性ベース署名と考えられており、より多くの構成方法が知られている。

Maji ら [1] により初めて属性ベース署名が提案されて以降、扱えるアクセス構造をより豊かにする研究が進み、今では論理回路やチューリング機械によって表現できる非常に幅広いアクセス構造を扱える属性ベース署名が知られている [2, 3, 4]。いずれの方式も構成方法の概略は似ている。以下典型的な、論理回路をアクセス構造に扱える平文ポリシー型の属性ベース署名の実現方法の概略を説明する。記載する方式は、初めて論理回路をアクセス構造に扱える方式を楢門曲線上で達成した坂井ら [2] の方式を、電子署名・コミットメント方式・非対話型ゼロ知識証明と呼ばれる暗号要素技術で一般化した El Kaafarani と勝又 [4] の方式に基づいている。

■論理回路をアクセス構造に扱える平文ポリシー型の属性ベース署名

Setup: 管理者は、Setup アルゴリズムを実行し、システム共有のパラメータを生成する。具体的に、電子署名の公開鍵 vk と秘密鍵 sk 、コミットメント方式の公開鍵 pk_c を生成し、属性ベース署名の管理者公開鍵 $mpk = (vk, pk_c)$ を公開し、管理者秘密鍵 $msk = sk$ を保管する。

KeyGen(msk, x): 管理者は、KeyGen アルゴリズムを実行し、(L ビットにエンコードされた) 属性 $x \in \{0, 1\}^L$ を所持するユーザの署名鍵 K_x を生成する。具体的に、管理者は sk を用いて、 x に対して署名 σ を生成し、署名鍵 $K_x = \sigma$ を

出力する。

Sign(mpk, K_x , C , msg): 属性 x を所持するユーザは、**Sign** アルゴリズムを実行し、 $C(x) = 1$ を満たす論理回路 $C : \{0, 1\}^L \rightarrow \{0, 1\}$ で表されるアクセス構造が付随する平文 msg に対して署名を生成する。具体的に、以下の手順で署名を生成する。

1. 属性 $x \in \{0, 1\}$ と署名鍵 $K_x = \sigma$ を管理者公開鍵に含まれる pk_c でコミットし、それらを各々 com_x と com_σ と置く。
2. 属性 $x \in \{0, 1\}$ と署名鍵 $K_x = \sigma$ を秘匿にしながら、次のような関係を示す非対話型ゼロ知識証明 π を生成し、 $\Sigma = (com_x, com_\sigma, \pi)$ を署名として出力する。本説明では、平文 msg が署名 Σ に紐づいていないが、一般的には、論理回路 C か非対話型ゼロ知識証明の内部に msg を埋め込むことで紐づけられることに注意されたい。
 - (a) $C(x) = 1$ である。つまり、属性がアクセス構造を満たす。
 - (b) σ が管理者公開鍵に含まれる vk の下で正当な署名である。
 - (c) com_x と com_σ が管理者公開鍵に含まれる pk_c の下で x と σ に対するコミットメントである。

Verify(mpk, Σ , C , msg): 検証者は、**Verify** アルゴリズムを実行し、論理式 C によって表されるアクセス構造を満たす属性を所持したユーザが msg に対して署名 Σ を生成したかを確認する。具体的に、非対話型ゼロ知識証明の証明 Σ の第 3 項 π が正しいことを検証する。

■**安全性について** 属性ベース署名は、適応的文書攻撃に対して存在的偽造不可能性 (EUF-CMA 安全性)、および、秘匿性を満たす必要がある。前者は、一般的な電子署名における EUF-CMA 安全性に似ており、攻撃者がある論理回路 C^* で表されるアクセス構造が付随した（以下、 C^* が付随した）平文 msg に対する署名 Σ を偽造できないことを要求する。この際、攻撃者は、 $C^*(x) = 0$ を満たす任意の属性 x （つまり、アクセス構造を満たさない x ）に対する署名鍵 K_x を管理者から入手でき、さらに、異なる論理回路 $C \neq C^*$ が付随した平文 msg に対する署名を得られる。後者は、属性ベース署名に特有な安全性要件で、署名から属性情報が漏れないことを要求する。具体的には、異なる属性 x^* と $x^{*'}$ が論理回路 C^* を満たす場合、 C^* が付随した平文 msg に対して属性 x と $x^{*'}$ を所持するいずれのユーザが署名を打ったかが識別できないことを要求する。これは、署名からはアクセス構造を満たした誰かが署名を作成したことしか漏れず、具体的な属性が秘匿されることを表す。

上記で紹介した方式の EUF-CMA 安全性は、概略的には電子署名の EUF-CMA 安全性と非対話型ゼロ知識証明の健全性から導かれる。仮に攻撃者に属性ベース署名の偽造ができたとした場合、非対話型ゼロ知識証明の健全性より、管理者公開鍵に含まれる電子署名の公開鍵 vk の下で正当な署名 σ を利用していなければならない。攻撃者は、管理者から $C^*(x) = 0$ を満たす任意の属性 x に対する署名鍵 $K_x = \sigma'$ を入手できるが、これらは C^* が付随した平文に署名することはできない。従って、攻撃者が偽造に利用した σ は、管理者から入手した σ' とは異なるが、これは電子署名の EUF-CMA 安全性に矛盾する。以上より、属性ベース署名の偽造を生成できる攻撃者がいないことが示される。

また、秘匿性については、コミットメントの秘匿性と非対話型ゼロ知識証明のゼロ知識性から直ちに導かれる。署名 Σ は、 (com_x, com_σ, π) という形をしているが、コミットメントの秘匿性より com_x と com_σ は、属性 x と署名鍵 K_x の情報を隠す。さらに、非対話型ゼロ知識証明のゼロ知識性より、証明 π も正しい証拠を利用して証明が作られた以上の情報を漏らさない。以上より、 Σ は、論理回路 C^* を満たす属性で生成された署名である以上の情報を漏らしていない。

■具体的な実現方法について 現在知られている属性ベース署名のほとんどは上記構成に従っており、違いはどのような電子署名、コミットメント、非対話型ゼロ知識証明を用いるかである。例えば、坂井ら [2] は上記一般構成の 2.(a) と 2.(b), (c) に各々特化した非対話型ゼロ知識証明 Groth-Sahai [5] と Groth-Ostrovsky-Sahai [6] を融合させることにより、Pairing 写像に基づく効率的な属性ベース署名を構成した。一方で、El Kaafarani と勝又 [4] は上記の暗号要素技術を耐量子計算機安全な道具で実現することで、初めて耐量子計算機安全な論理回路をアクセス構造に扱える属性ベース署名を構成した。

3.2.2 集約 MAC、マルチ MAC、集約署名、マルチ署名

メッセージ認証コード (Message Authentication Code, MAC) や電子署名は、ある平文を特定のユーザが認証したこと、署名したことを暗号的に保証できる暗号技術である。MAC は秘密鍵を利用してタグを検証することができ、電子署名は公開情報のみで署名を検証できる。通常の MAC や電子署名では、複数のユーザや平文に対してタグを付与したり、署名を生成する場合、MAC や署名を行う回数だけタグや署名が生成されるため、タグや署名が数多く生成される。アプリケーションにおいてはスケーラビリティの課題となる。そこで、この課題を解決するための技術である集約 MAC、マルチ MAC、集約署名、マルチ署名を紹介する。

- **集約 MAC** [7]: 異なる平文に対して、複数の異なる秘密鍵で生成されたタグを、単一のタグに集約する技術であり、タグサイズの削減が可能となる。タグ検証においても、集約したタグの検証のみで複数の平文の正当性を一括で検証可能となる。ただし、平文や秘密鍵は必ずしも異なるとは限らない。
- **マルチ MAC** [7]: 単一の平文に対して、複数の異なる秘密鍵で生成されたタグを、単一のタグに集約する技術である。タグ生成の際の平文が全ユーザで同一な、集約 MAC の特殊ケースとして捉えることができ、集約 MAC と比べてより効率的な構成が可能となる。
- **集約署名** [8, 9]: 電子署名の枠組みで集約 MAC と同じ機能を実現する方法である。
- **マルチ署名** [8, 9, 10, 11, 12]: 電子署名の枠組みでマルチ MAC と同じ機能を実現する方法である。

集約・マルチ MAC/署名の中には、集約する際に MAC/署名を生成するユーザ同士が対話する必要がある方式や、集約が逐次的にしか行えない方式等様々な方式があるため、用途に応じて選定することが求められる。一般的に、利便性が高い方式ほど効率性が悪くなる傾向があることに注意されたい。例えば、常にネットに繋がっているサーバ同士でタグや署名を集約する場合は、効率を高めるために対話型の集約方式の利用が好ましい。一方で、一般のユーザがクラウドやブロックチェーン上にアップロードしたタグや署名を集約する応用では、全ユーザが同時にオンラインであるとは限らないため、非対話型の集約方式の方が好ましい。

以下、代表的な方式を二つを紹介する。

■Katz と Lindell による集約・マルチ MAC [7] F を鍵空間と出力空間が $\{0, 1\}^\lambda$ で入力空間が $\mathbb{M}$ の擬似ランダム関数とする。以下、平文空間に $\mathbb{M}$ を持つ集約 MAC を F から構成する。

KeyGen(1^λ): 各ユーザは、鍵生成アルゴリズムを実行し、 F の鍵 $K \leftarrow \{0, 1\}^\lambda$ をサンプルし、秘密鍵 $\text{key} = K$ を出力する。

MAC($\text{key}, \text{msg} \in \mathbb{M}$): ユーザは、タグ生成アルゴリズムを実行し、タグ $\text{tag} \leftarrow F(K, \text{msg})$ を生成する。

Aggregate($((\text{msg}_i, \text{tag}_i)_{i \in [N]})$): タグ集約者は、各ユーザが生成した平文 msg_i に対するタグ tag_i を入力にタグ集約アルゴリズムを実行し、集約タグ $\text{tag} = \bigoplus_{i \in [N]} \text{tag}_i \in \{0, 1\}^\lambda$ を生成する。

Verify($((\text{key}_i, \text{msg}_i)_{i \in [N]}, \text{tag})$): タグ検証者は、各ユーザの秘密鍵 $\text{key}_i = K_i$ と署名したであろう平文 msg_i を入力にタグ検証アルゴリズムを実行し、 $\text{tag} =$

$\bigoplus_{i \in [N]} F(K_i, \text{msg}_i)$ が成り立てば、集約タグが正当だと判定する。ただし、異なる $i, j \in [N]$ に対して $(\text{key}_i, \text{msg}_i) \neq (\text{key}_j, \text{msg}_j)$ とする。

タグ検証アルゴリズムはタグ生成を再度行い、それが集約タグと同一かを確認する。タグ生成が決定的であることより、方式の正当性は確認できる。参考までに、上記方式は擬似ランダム関数の代わりに、タグ生成が決定的な任意の MAC を利用することができる。詳細は [7] を参考にされたい。また、上記方式はタグ集約アルゴリズムが同じ平文だけを入力に取るように制限することで、マルチ MAC にもなる。

■集約・マルチ MAC の安全性について 通常の MAC と同様に、適応的平文と署名攻撃に対して存在的偽造不可能性 (EUF-CMVA 安全性) と同等の安全性を要求する。この安全性は、秘密鍵 $\text{key}_i = K_i$ を持つユーザが平文 msg_i を署名していない限り、 $\text{Verify}((\text{key}_i, \text{msg}_i)_{i \in [N]}, \text{tag})$ が正当と判断するような組み $(\text{key}_i, \text{msg}_i)_{i \neq j, i \in [N]}$ と集約タグ τ を攻撃者が出力できないことを意味する。ただし、攻撃者は任意のタグと平文の組み (σ, msg) が秘密鍵 key_i の元で正しい署名か検証してもらえとするとする。EUF-CMVA 安全性を上記方式が満たすことは、 F が擬似ランダム関数であることより直ちに導かれる。

■Boneh, Drijvers と Neven による集約・マルチ署名 [9] $\mathbb{G}_0, \mathbb{G}_1, \mathbb{G}_T$ を素数位数 p を持つ群として、 g_0 を $\mathbb{G}_0$ の生成元、 $e : \mathbb{G}_0 \times \mathbb{G}_1 \rightarrow \mathbb{G}_T$ を付随する (非対称) Pairing 写像とする。また、 H_0 と H_1 をハッシュ関数とする。以下、平文空間に $\mathbb{M}$ を持つ集約署名を構成する。

KeyGen : 各ユーザは、鍵生成アルゴリズムを実行し、 $a \leftarrow \mathbb{Z}_p$ をサンプルし、公開鍵 $\text{pk} = h = g_0^a \in \mathbb{G}_0$ と秘密鍵 $\text{sk} = a$ を生成する。

Sign(sk, msg) : ユーザは、署名生成アルゴリズムを実行し、署名 $\sigma = H_0(\text{msg})^{\text{sk}} \in \mathbb{G}_1$ を生成する。

Aggregate($((\text{pk}_i, \text{msg}_i, \sigma_i)_{i \in [N]})$) : 署名集約者は、各ユーザが生成した平文 msg_i に対する署名 σ_i を入力に署名集約アルゴリズムを実行し、まず $(t_1, \dots, t_N) \leftarrow H_1(\text{pk}_1, \dots, \text{pk}_N) \in \mathbb{Z}_p^N$ を計算し、集約署名 $\sigma = \prod_{i \in [N]} \sigma_i^{t_i}$ を生成する。

Verify($((\text{pk}_i, \text{msg}_i)_{i \in [N]}, \sigma)$) : 署名検証者は、各ユーザの公開鍵 $\text{pk}_i = g_0^{a_i}$ と署名したであろう平文 msg_i を入力に署名検証アルゴリズムを実行し、まず $(t_1, \dots, t_N) \leftarrow H_1(\text{pk}_1, \dots, \text{pk}_N) \in \mathbb{Z}_p^N$ を計算し、 $e(g, \sigma) = \prod_{i \in [N]} e(\text{pk}_i^{t_i}, H_0(\text{msg}_i))$ が成り立てば、集約タグが正当なものだと判定する。

アルゴリズムの正当性は、下記の数式が成立するこから確認できる。

$$e(g_0, \sigma) = e(g_0, \prod_{i \in [N]} \sigma_i^{t_i}) = \prod_{i \in [N]} e(g_0, H_0(\text{msg}_i)^{\text{sk}_i \cdot t_i}) = \prod_{i \in [N]} e(\text{pk}_i^{t_i}, H_0(\text{msg}_i))$$

さらに、各ユーザが同じ平文 msg を署名するマルチ署名の設定においては、予め集約された公開鍵 $\text{apk} = \prod_{i \in [N]} \text{pk}_i \in \mathbb{G}_0$ を計算することで、集約署名の検証は $e(g, \sigma) = e(\text{apk}, H_0(\text{msg}))$ を確かめるだけである。計算コストが比較的に高い Pairing 写像の計算が $(n + 1)$ 回から 2 回にでき、マルチ署名の方が集約署名よりも検証コストが少なくなる。集約・マルチ MAC の場合、このような検証コストの差はない。

■集約・マルチ署名の安全性について 集約・マルチ MAC の場合と同様、通常の電子署名における適応的平文攻撃に対して存在的偽造不可能性 (EUF-CMA 安全性) と同等の安全性を要求する。ただし、MAC とは違い、電子署名では検証を攻撃者自身が行えるため、選択署名攻撃を考慮に入れなくてよいことに注意されたい。上述の方式は、Boneh ら [13] の電子署名の自然な集約・マルチ署名への拡張になっているが、EUF-CMA 安全性も大まかには同様に導かれる。具体的には、 $(g_0^a, g_1^a, g^b) \in \mathbb{G}_0 \times \mathbb{G}_1 \times \mathbb{G}_1$ を与えられても $g_0^{ab} \in \mathbb{G}_0$ を求めることができないという、co-Computational Diffie-Hellman 仮定の困難性より導かれる。

■その他の方式 署名の設定において、現在最も効率的な集約署名は上記で紹介した Boneh ら [9] の方式である。マルチ署名については、署名集約アルゴリズムで各ユーザが対話することを許せば、Pairing 写像を用いないより効率的な方式がある。Bellare と Neven[10] は、対話を 3 ラウンド許すことで離散対数問題の困難性仮定を基にしたマルチ署名を構成し、署名長は Schnorr 署名 [14] と同じである。近年、より強い困難性仮定を基にすることで、[10] の方式のラウンド数を減らしより効率性を高めた方式が提案されている [11, 12]。

3.2.3 グループ署名

■グループ署名の特徴 従来の電子署名では、署名検証の際に署名者が誰であるか明示的に分かってしまう。グループ署名は、従来の電子署名と異なり、署名者の匿名性を保証することができる特徴を有する。グループ署名における匿名性とは、ある組織 (グループ) に属するメンバのだれかが署名したことしか検証者には分からない性質である。グループ署名では、署名者と検証者に加えて、発行者と呼ばれる各グループメンバに署名生成用の署名鍵を発行する管理者と、開示者と呼ばれる署名から真の署名者を開示する管理者を含む。ただし、それぞれの管理者は機

能分離されていなくてもよい。

署名者の匿名性を保証する高機能暗号技術としてリング署名 (第 3.2.4 章参照) があるが、グループ署名とリング署名は以下の相違がある。

- アドホック性：リング署名では各署名者は独自に鍵生成を行い検証鍵を公開し、署名生成時に任意の検証鍵集合を選択して含めることでグループをアドホックに構成できるのに対し、グループ署名では発行者と署名鍵を生成したユーザ集合がグループとなる。
- 署名者開示の有無：リング署名では署名鍵が開示されたとしても真の署名者であるかどうか分からないのに対し、グループ署名では開示者により真の署名者を開示できる。

この相違に応じ、リング署名は管理者無しの環境で強い匿名性を要求する状況、グループ署名は管理者がグループを管理し緊急時には真の署名者を開示可能であることを要求する状況で使い分けることが望ましい。

■グループ署名の歴史 グループ署名の概念は、Chaum と Van Heyst [15] によって初めて提案された。同時にいくつかの方式が提案されたが、グループメンバをセットアップ後は追加不可能である静的グループに制限されている方式であるか、もしくは開示者が真の署名者を開示する際にグループメンバとの対話が必要という問題があった。また、数学的な安全性定義が示されておらず、非形式的な安全性評価しかされていなかった。

その後、Bellare らによって、静的グループの場合 [16] と署名生成後でも新たなグループメンバを追加可能な動的グループの場合 [17] における統一的な数学的安全性定義がそれぞれ提案された。加えて、Bellare らは従来の電子署名、公開鍵暗号、非対話ゼロ知識証明から一般的に安全なグループ署名を構成する手法を提案した。Bellare らの構成法では、メンバ追加時は、各メンバは発行者から自分がグループに属していることを示す証明書を受け取り、証明書を含む署名鍵を生成する。署名生成時は、自分の ID に紐づいた証明書の暗号文とともに、証明書を知ることと暗号文が正しいことを証明する非対話ゼロ知識証明を署名として出力する。開示者による署名開示時は、署名に含まれる暗号文を復号し、ID を特定した上で、(必要に応じて) 開示内容が正しいことを示す非対話ゼロ知識証明を出力する。Bellare らの安全性定義に基づき、Pairing 写像を利用した署名長の短い方式 [18, 19] 等が提案されている。

また、グループメンバの証明書を失効する機能を持つ方式も知られており、グループ管理者が失効リストを配布し、署名者がそれに基づいて署名を生成し、検証

者が失効済みかどうか検証するリスト公開型失効 [20] と、署名生成時は失効リストを用いずに検証者のみがリスト検証する検証者ローカル型失効 [21] がある。検証者ローカル型失効では、新たな失効者が出る際に、その都度、グループメンバに通知する必要があるため、リスト公開型失効よりも効率的である。

近年では、量子計算機実用化の研究の進展に伴い、大規模な量子計算機が実現したとしても安全性を保つ耐量子計算機安全な方式の研究が進んでおり、格子問題に基づく方式 [22, 23, 24] が主に提案されている。

■**グループ署名方式の比較** 一般に、グループ署名の安全性として以下の 3 つの性質が要求される。

- 匿名性：開示者以外（発行者も含む）は署名から真の署名者を識別できない。
- 追跡可能性：開示者が署名者を追跡不可能であり、かつ、検証に合格する署名を攻撃者は生成することができない。
- 濡れ衣不可能性：管理者であっても、署名を作成していないメンバに対して真の署名者であると証明できない。

Bellare らの構成法 [16, 17] 以降に提案された方式の多くは、彼らの構成法を基としており、上記の性質を満たしている。

また、グループ署名の効率は以下の観点で比較できる。

- グループ公開鍵長
- 署名長
- 署名計算量

Chaum と Van Heyst の方式 [15] の方式は、グループ公開鍵長、署名長、署名計算量のいずれもグループメンバ数に比例しているが、Camenisch と Stadler [25] によってグループ公開鍵長、署名長、署名計算量のいずれも定数長の RSA 仮定に基づく方式が提案され、以後古典計算機に対して安全な方式のほとんどは定数長を達成している。その中でも、Pairing 写像を利用したグループ署名方式 [18, 19] では署名長をグループ署名ではない CRYPTREC 暗号リスト [26] に掲載されている電子署名の数倍程度まで短くすることができる。耐量子計算機安全な方式では、グループ公開鍵長と署名計算量が定数長の方式 [24] やグループ公開鍵長と署名長が定数長の方式 [23] が知られている。

■**代表的な方式のアルゴリズム** 具体的なグループ署名の実現方法として、公開鍵暗号と電子署名に基づく静的な方式 [16] を紹介する。

Setup($1^k, 1^n$): 管理者鍵生成アルゴリズム。 k をセキュリティパラメータ、 n をグループの最大メンバ数とし、共通参照情報 crs 、公開鍵暗号の復号鍵と暗号化鍵 (dk, ek) 、電子署名の署名鍵と検証鍵 (msk, mvk) を生成し、発行者の秘密鍵 $ik = msk$ 、開示者の秘密鍵 $ok = dk$ 、グループ公開鍵 $gpk = (crs, mvk, ek)$ を出力する。

KeyGen(i, ik, gpk): メンバ鍵生成アルゴリズム。メンバ i は電子署名の署名鍵と検証鍵 (sk_i, vk_i) を生成し、 vk_i を発行者に送る。発行者は msk を用いて、 (i, vk_i) をメッセージとした署名 $cert_i$ を生成し、証明書としてメンバに送る。メンバは秘密鍵 $gsk_i = (i, sk_i, vk_i, cert_i)$ を出力する。

Sign(gsk_i, m): 署名者 i の署名生成アルゴリズム。メッセージ m に対して、 sk_i を用いて署名 s を生成し、乱数 r を選び、 ek を用いて $(i, vk_i, cert_i, s)$ を暗号化した暗号文 ct を計算する。 crs を用いて、次のような関係を示す非対話ゼロ知識証明 π を生成する。 (mvk の下で $((i, vk_i), cert_i)$ が正しい署名) かつ (vk_i の下で (m, s) が正しい署名) かつ (ek の下で ct が $(i, vk_i, cert_i, s)$ の正しい暗号文) 署名 $\sigma = (ct, \pi)$ を出力する。

Verify(gpk, m, σ): 署名検証アルゴリズム。 π が成り立つかチェックする。

Open(ok, gpk, m, σ): 署名者開示アルゴリズム。 π が成り立つかチェックし、 dk を用いて ct から $(i, vk_i, cert_i, s)$ を復号し、 i を出力する。

3.2.4 リング署名

■**リング署名の特徴** リング署名は、匿名署名の一種であり、あるグループの誰かが署名したことはわかる（偽造不可能性）が、そのグループの誰が署名したかはわからない（匿名性）、という性質を持っている。具体的には、リング署名は、リングと呼ばれる n 人の署名参加者の集合に対して、リングのなかの少なくとも 1 人の署名者が署名を生成したことは確認できるが、リングのなかの誰が署名を生成したかはわからない、ことを示す署名である。リング署名は、リングに属する n 人の署名参加者の公開鍵に対して、少なくとも 1 人の署名参加者の秘密鍵を知っていることを示す、非対話 1-out-of- n OR 証明と見ることもできる。

匿名署名の一種であるグループ署名（第 3.2.3 章参照）と比較すると、以下のような相違がある。

- **アドホック性：**リング署名では、各署名者は独自に鍵生成を行い検証鍵を公開し、署名生成時に署名者が任意の検証鍵集合を選択することでグループをアドホックに構成できる。一方、グループ署名では、発行者（グループ管理

者)と協力して署名鍵を生成したユーザの集合がグループとなる。

- 署名者開示の有無：リング署名では、署名者であっても、真の署名者を開示することはできない。一方、グループ署名では、開示者（グループ管理者）は、真の署名者を開示することができる。

このように、リング署名は、署名生成時にアドホックにグループを選ぶことができ、グループ管理者が必要なく非中央集権的である、という特徴を持っている。

■**リング署名の歴史** リング署名は、Rivest, Shamir, Tauman [27] によって初めて提案され、落戸付一方向性置換に基づく方式が示された。その後、離散対数問題に基づくリング署名方式 [28] が提案された。これらの方式で用いられている典型的なリング署名の構成法では、 n 人の署名参加者の集合であるリングに対して、リングのなかの署名者以外の署名参加者の公開鍵を用いて $n - 1$ 個の値の連鎖を生成し、署名者の秘密鍵を用いて連鎖を環になるように連結する 1 個の値を生成することによって、値の連鎖の環を構成しそれを署名としている。署名者の秘密鍵を用いて連結した部分は、署名者の秘密鍵が分かったとしても特定することができず、強い匿名性を実現している。また、署名サイズは $O(n)$ となっている。

通常のリング署名は前述の通り、署名者であっても自分が真の署名者であることを証明できない強い匿名性を保証しているが、応用のために匿名性を制限した方式も知られている。リンク可能リング署名 [29] では、同じ秘密鍵とタグを用いて異なる 2 つのメッセージについて生成された署名を、リンクすることができる。トレース可能リング署名 [30] では、同じ秘密鍵とタグを用いて異なる 2 つのメッセージについて生成された署名から、署名者を特定することができる。リング署名の非中央集権的な構成が暗号資産に適しており、上記の性質により暗号資産の二重使用を防ぐことができるため、リンク可能リング署名やトレース可能リング署名は匿名の暗号資産で用いられている。

また、しきい値リング署名 [31, 32] は、1-out-of- n のリング署名を k -out-of- n に一般化したものであり、 n 人の署名参加者の集合であるリングに対して、リングのなかの少なくとも k 人の署名者が協力して署名を生成したことを確認することができる署名である。

■**リング署名方式の比較** リング署名の安全性として以下の 2 つの性質が要求される。

- 匿名性: n 人のリングのなかの誰が署名者であるか、署名から特定できない。
- 偽造不可能性: n 人のリングのなかの少なくとも 1 人の署名者の秘密鍵がなければ、署名を生成できない。

リング署名方式の多くは、上記の性質を満たしている。匿名性を制限したものとして、前述のリンク可能リング署名とトレース可能リング署名がある。さらに、アカウント可能リング署名 [33] では、開示者を置くことによって署名者の開示を可能としている。ただし、アカウント可能リング署名では、グループ署名とは異なり発行者は必要なく署名生成時にグループを選ぶことができる。また、偽造不可能性を k -out-of- n に一般化したものとして、前述のしきい値リング署名がある。

また、リング署名の効率は以下の観点で評価される。

- 署名サイズ
- 署名生成と検証に必要な計算量

リング署名方式の多くは、リングの署名参加者の数を n とすると、署名サイズが $\mathcal{O}(n)$ である。1-out-of- n の効率的なゼロ知識証明を用いることで、署名サイズを $\mathcal{O}(\log n)$ にできる方式が知られている [34, 35, 36, 37]。また、信頼できるセットアップを仮定することで、アキュムレータにより n 個の公開鍵を圧縮し署名サイズを定数にできる方式も提案されている [38]。

また、基づいている計算量仮定の観点では、耐量子計算機安全性を持つリング署名方式が提案されている [39, 35, 36, 37]。

■代表的な方式のアルゴリズム 具体的なリング署名の実現方法として、Abe, Ohkubo, Suzuki による離散対数問題に基づく方式 [28] を紹介する。署名者 i について、 q_i を素数、 g_i を位数 q_i の巡回群 G_i の生成元、 $H_i : \{0, 1\}^* \rightarrow \mathbb{Z}_{q_i}$ をハッシュ関数とする。

KeyGen(q_i, G_i, g_i): 鍵生成アルゴリズム。

1. 署名者 i は、秘密鍵 $x_i \in \mathbb{Z}_{q_i}$ をランダムに選び、公開鍵 $(q_i, G_i, g_i, y_i = g_i^{x_i})$ を計算する。

Sign($x_a, L = \{y_1, \dots, y_n\}, m$): 署名生成アルゴリズム。

1. 署名者 $a \in \{1, \dots, n\}$ は、秘密鍵 x_a と公開鍵のリスト $L = \{y_1, \dots, y_n\}$ とメッセージ m を入力として受け取る。
2. 署名者 a は、 $r_a \in \mathbb{Z}_{q_a}$ をランダムに選び、 $e_a = g_a^{r_a}$ と $c_{a+1} = H_{a+1}(L, m, e_a)$ を計算する。
3. 署名者 a は、 $s_i \in \mathbb{Z}_{q_i}$ をランダムに選び、 $e_i = g_i^{s_i} y_i^{c_i}$ と $c_{i+1} = H_{i+1}(L, m, e_i)$ を、 $i = a + 1, \dots, n, 1, \dots, a - 1$ (ただし $i = n + 1 = 1$ とする) について計算する。
4. 署名者 a は、 $s_a = r_a - c_a x_a \pmod{q_a}$ を計算する。

- 署名者 a は、署名 $\sigma = (c_1, s_1, \dots, s_n)$ を出力する。

Verify($L = \{y_1, \dots, y_n\}, \sigma, m$): 署名検証アルゴリズム。

- 検証者は、公開鍵のリスト $L = \{y_1, \dots, y_n\}$ と署名 $\sigma = (c_1, s_1, \dots, s_n)$ とメッセージ m を入力として受け取る。
- 検証者は、 $e_i = g_i^{s_i} y_i^{c_i}$ と $c_{i+1} = H_{i+1}(L, m, e_i)$ を、 $i = 1, \dots, n-1$ について計算する。
- 検証者は、 $e_n = g_n^{s_n} y_n^{c_n}$ を計算する。
- 検証者は、もし $c_1 = H_1(L, m, e_n)$ なら受理を出力し、そうでなければ非受理を出力する。

なお、この方式は、 $n = 1$ のとき Schnorr 署名となっており、Schnorr 署名の自然な拡張になっている。

3.2.5 しきい値署名

■しきい値署名の特徴 (k, n) しきい値署名とは、ある $(k, n : k \leq n)$ である自然数の組に対して、以下のアルゴリズムから成る公開鍵署名方式である。

- 公開鍵署名の署名生成鍵と検証鍵のペアを生成する。
- 署名生成鍵を n 個のシェアに分割する（しきい値秘密分散）。シェアとは、元のデータを複数のデータに分割したものである。
- 1つのメッセージに対していくつかのシェアで署名を生成する。
- シェアから生成された署名を k 個以上集め、組み合わせる処理を行う。
- 組み合わせられた署名に対し、検証鍵で検証を行い、検証結果を出力する。

偽造等の不正が行われず正当にこのアルゴリズムが実行された場合、第4項で組み合わせられた署名は、第1項で生成された署名鍵を用いて生成された署名と同等となり、第5項の検証処理を通る署名となる。

しきい値署名は一般的な DSA 等の公開鍵署名方式と比較すると、ベースとなるしきい値秘密分散の特性によって署名生成鍵を複数にすることができるため、しきい値未満の漏洩や故障に対して優位性を持つ。類似したアクセス構造を実現する署名方式として「署名生成の際に設定された（しきい値等の）アクセス構造を満たす属性集合に対応する署名生成鍵を持つ署名者は単一の正当な署名を生成できる」属性ベース署名が存在するが、属性ベース署名では署名者は単一であり、単体ではアクセス構造を満たさない属性を複数の署名者が持ち寄っても正当な署名は作れない。一方でしきい値署名は複数の署名者に署名生成権限を分散すること

を想定しており、漏洩・故障等への耐性を持つことが属性ベース署名とは異なる。また、マルチパーティ計算のように複数の参加者が想定される署名プロトコルであるため、鍵生成については、信頼できる鍵生成・配布を行うトラステッドディーラーを仮定する場合と、各参加者が独自に鍵を生成した上で暗黙的に署名生成鍵や署名検証鍵が決定される分散鍵生成を用いる方式が存在する。

しきい値署名におけるの安全性の定義には以下が存在する。

- 偽造不可能性 (Unforgeability)：しきい値未満のシェアで生成された署名を集めただけでは、検証鍵で検証可能である正当な署名は生成不可能。さらに不正者の数によって以下のように定義が分かれる。
 - dishonest majority：不正者が $n/2$ 以上しきい値未満
 - honest majority：不正者が $n/2$ 、かつしきい値未満
- 堅牢性 (Robustness)：正規のシェアがしきい値以上集まれば、必ず正当な署名が生成可能
- Identifiable abort：正直な署名者は不正なシェアが少なくとも 1 つ以上含まれていることを検知可能
- Proactive security：長期的な期間でのシェアの漏洩に対して安全

また署名アルゴリズムとしての安全性もあるため、例えば後述する BLS 署名をベースとしたしきい値署名ではランダムオラクルモデルのもとで Gap Diffie-Hellman 問題の困難性を安全性の根拠として、選択メッセージ攻撃に対する存在的偽造不可能性が証明されている。

■しきい値署名の歴史 しきい値署名は 1987 年 Desmedt によって最初にコンセプトが提案 [40] され、同じく Desmedt と Frankel によって ElGamal 暗号 [41] と Shamir のしきい値秘密分散 [42] を組み合わせた方式が提案された [43]。Gennaro らは安全性証明がついた ElGamal 方式ベースのしきい値署名を 1996 年に発表し [44]、Shoup は同期通信を必要としない RSA 署名をベースとした方式を発表している [45]。また署名の検証に署名生成者の協力が必要な Undeniable な方式や、検証者を複数にする (k, l) -Shared Verification 方式等、様々な方式が提案されている。

近年では暗号資産に関連した技術として注目されている。例えば、Bitcoin では署名者のプライバシーに関する問題やトランザクションのストレージ効率化に向けてしきい値署名を採用しやすい Schnorr 署名をとりいれている [46]。また、効率的なスキームとして FROST [47] 等が考案されており、利用されるケースが増えてきている。ECDSA を取り入れたしきい値署名に関する提案が数多くあり [48] セ

セキュリティの強化やプロトコルの効率化を目指して様々な方式があげられている [49, 50, 51, 52, 53, 54, 55, 56]。最近では Groth と Shoup によって効率的で堅牢な方式 [57] が提案されている。

■しきい値署名方式の比較 しきい値署名は様々な方式が提案されているがベースとなる署名方式やしきい値秘密分散・分散鍵生成の方式によって効率性や安全性が異なる。

効率性では、(1) 鍵長・署名長などの各パラメータサイズ、(2) シェア生成・組み合わせ処理などの各オペレーション速度、(3) エンティティ間の同期が必要か否かの違いがあげられるが、(1)(2) はベースとなる署名方式や定義されている演算処理（楕円曲線演算や Pairing 写像）によって異なる。例えば、128bit セキュリティ程度のパラメータを基準とすると ECDSA (secp256r1) および EdDSA (Ed25519) ベースのしきい値署名 [57, 47] の各シェア自身のサイズは署名生成鍵と同等で 256bit、シェアから生成される署名長は 512bit となる。BLS 署名 (BLS12-381) ベースのしきい値署名 [58] の場合、各シェアのサイズは 256bit、署名長は 768bit となる [59]。なお、数値については圧縮を使った表現も含まれており、シェアに対するコミットメントの保持や通信回数の効率化や安全性の向上のために事前計算を行った signature helpers を共有することもあるため、参加者が保持するデータ量はこれより増加することがある。(3) については方式が同期式・非同期式なのかによって異なり先の項目でもあげたが事前計算を行うことによって非同期式を実現し高速化を行うことがある。また鍵生成方式においてもベースとなる分散鍵生成方式の違いによって決まってくる。例えば通信ラウンド数の比較では Pedersen-DKG[60] は 3 ラウンド、Gennaro-DKG[61] は 4 ラウンド、FROST-DKG[47] は 1 ラウンドというように方式によって大きな違いがありネットワークレイテンシを考慮すると性能に大きな差が生じる可能性がある。しきい値署名に関する各文献で多くの方式が比較されているが、最近の比較では [56] の Figure1 や [62] の Table1 などがあげられる。

■代表的なアルゴリズム・方式 ここでは Ethereum など既に用いられている Boldyreva が提案した BLS 署名 [13] ベースのしきい値署名 [58] を紹介する。以下では、 (t, n) -しきい値署名とし、 n 人の参加者を $P_1, \dots, P_n$ とする

Setup(1^κ): BLS 署名のセットアップはセキュリティパラメータ κ を入力にとり、素数 p 、 p を位数とする巡回群 G_1, G_2, G_T 、各巡回群を使って定義される Pairing 写像 $e : G_1 \times G_2 \rightarrow G_T$ 、文字列から G_1 の元へのハッシュ関数 $H : \{0, 1\}^* \rightarrow G_1$ 、 G_2 の元からランダムに決定した Q を準備し公開パラ

メータ $Params := (p, G_1, G_2, G_T, e, H, Q)$ とする

$SharedKeygen(Params, t, n)$: 分散鍵生成アルゴリズム [61] を利用して BLS 署名鍵ペア (vk, sk) (ただし、 $sk \in \mathbb{Z}_p^*, vk = skQ$) に対するシェア $\{vk_i, sk_i : 1 \leq i \leq n\}$ を (t, n) 秘密分散になるよう各参加者が保持する。

$SharedSign(Params, m, sk_i)$: 参加者はメッセージ m に対して $\sigma_i = sk_i H(m)$ を計算し提出する

$Verify((Params, m, \{\sigma_{i_j}\}_{1 \leq j \leq t}, \{vk_{i_j}\}_{1 \leq j \leq t}, vk))$: 1. 参加者から収集した各 j に対して、 $e(\sigma_{i_j}, Q) = e(H(m), vk_{i_j})$ が成り立つことを確認する
2. 正規の署名を $\sigma = \sum_{1 \leq j \leq t} L(i_j) \sigma_{i_j}$ により計算する。ただし、 $L(i_j)$ は分散鍵生成におけるラグランジュ補間係数である。
3. vk を使って $e(\sigma, Q) = e(H(m), vk)$ により署名を検証する。

参考文献

- [1] H. K. Maji, M. Prabhakaran and M. Rosulek: “Attribute-Based Signatures”, CT-RSA 2011, Springer-Verlag, pp. 376–392 (2011).
- [2] Y. Sakai, N. Attrapadung and G. Hanaoka: “Attribute-Based Signatures for Circuits from Bilinear Map”, PKC 2016, Springer-Verlag, pp. 283–300 (2016).
- [3] Y. Sakai, S. Katsumata, N. Attrapadung and G. Hanaoka: “Attribute-Based Signatures for Unbounded Languages from Standard Assumptions”, ASIACRYPT 2018, Springer-Verlag, pp. 493–522 (2018).
- [4] A. El Kaafarani and S. Katsumata: “Attribute-Based Signatures for Unbounded Circuits in the ROM and Efficient Instantiations from Lattices”, PKC 2018, Springer-Verlag, pp. 89–119 (2018).
- [5] J. Groth and A. Sahai: “Efficient Noninteractive Proof Systems for Bilinear Groups”, SIAM J. Computing, **41**(5), pp. 1193–1232 (2012).
- [6] J. Groth, R. Ostrovsky and A. Sahai: “New Techniques for Noninteractive Zero-Knowledge”, J. ACM, **59**(3), pp. 1–35 (2012).
- [7] J. Katz and A. Y. Lindell: “Aggregate Message Authentication Codes”, CT-RSA 2008, Springer-Verlag, pp. 155–169 (2008).
- [8] D. Boneh, C. Gentry, B. Lynn and H. Shacham: “Aggregate and Verifiably Encrypted Signatures from Bilinear Maps”, EUROCRYPT 2003, Springer-Verlag, pp. 416–432 (2003).

- [9] D. Boneh, M. Drijvers and G. Neven: “Compact Multi-Signatures for Smaller Blockchains”, ASIACRYPT 2018, Springer-Verlag, pp. 435–464 (2018).
- [10] M. Bellare and G. Neven: “Multi-Signatures in the Plain Public-Key Model and a General Forking Lemma”, ACM CCS 2006, pp. 390–399 (2006).
- [11] C. Komlo and I. Goldberg: “FROST: Flexible Round-Optimized Schnorr Threshold Signatures”, SAC 2020, Springer-Verlag, pp. 34–65 (2020).
- [12] J. Nick, T. Ruffing and Y. Seurin: “MuSig2: Simple Two-Round Schnorr Multi-Signatures”, CRYPTO 2021, Springer-Verlag, pp. 189–221 (2021).
- [13] D. Boneh, B. Lynn and H. Shacham: “Short Signatures from the Weil Pairing”, J. of Cryptology, **17(4)**, pp. 297–319 (2004).
- [14] C.-P. Schnorr: “Efficient Identification and Signatures for Smart Cards”, CRYPTO 1989, Springer-Verlag, pp. 239–252 (1989).
- [15] D. Chaum and E. V. Heyst: “Group Signatures”, EUROCRYPT 1991, Springer-Verlag, pp. 257–265 (1991).
- [16] M. Bellare, D. Micciancio and B. Warinschi: “Foundations of Group Signatures: Formal Definitions, Simplified Requirements, and a Construction Based on General Assumptions”, EUROCRYPT 2003, Springer-Verlag, pp. 614–629 (2003).
- [17] M. Bellare, H. Shi and C. Zhang: “Foundations of Group Signatures: The Case of Dynamic Groups”, CT-RSA 2005, Springer-Verlag, pp. 136–153 (2005).
- [18] D. Boneh, X. Boyen and H. Shacham: “Short Group Signatures”, CRYPTO 2004, Springer-Verlag, pp. 41–55 (2004).
- [19] J. Groth: “Fully Anonymous Group Signatures without Random Oracles”, ASIACRYPT 2007, Springer-Verlag, pp. 164–180 (2007).
- [20] J. Camenisch and A. Lysyanskaya: “Dynamic Accumulators and Application to Efficient Revocation of Anonymous Credentials”, CRYPTO 2002, Springer-Verlag, pp. 61–76 (2002).
- [21] B. Boneh and H. Shacham: “Group Signatures with Verifier-Local Revocation”, ACM CCS 2004, pp. 168–177 (2004).
- [22] S. D. Gordon, J. Katz and V. Vaikuntanathan: “A Group Signature Scheme from Lattice Assumptions”, ASIACRYPT 2010, Springer-Verlag, pp. 395–412 (2010).

- [23] S. Katsumata and S. Yamada: “Group Signatures Without NIZK: From Lattices in the Standard Model”, EUROCRYPT 2019, Springer-Verlag, pp. 312–344 (2019).
- [24] V. Lyubashevsky, N. Nguyen, M. Plancon and G. Seiler: “Shorter Lattice-Based Group Signatures via “Almost Free” Encryption and Other Optimizations”, ASIACRYPT 2021, Springer-Verlag, pp. 218–248 (2021).
- [25] J. Camenisch and M. Stadler: “Efficient group signature schemes for large groups”, CRYPTO 1997, Springer-Verlag, pp. 410–424 (1997).
- [26] CRYPTREC: “電子政府における調達のために参照すべき暗号のリスト”.
<https://www.cryptrec.go.jp/list/cryptrec-ls-0001-2012r6.pdf>.
- [27] R. Rivest, A. Shamir and Y. Tauman: “How to Leak a Secret”, ASIACRYPT 2001, Springer-Verlag, pp. 552–565 (2001).
- [28] M. Abe, M. Ohkubo and K. Suzuki: “1-out-of-n Signatures from a Variety of Keys”, ASIACRYPT 2002, Springer-Verlag, pp. 415–432 (2002).
- [29] J. Liu, V. Wei and D. Wong: “Linkable Spontaneous Anonymous Group Signature for Ad Hoc Groups (Extended Abstract)”, ACISP 2004, Springer-Verlag, pp. 325–335 (2004).
- [30] E. Fujisaki and K. Suzuki: “Traceable Ring Signature”, PKC 2007, Springer-Verlag, pp. 181–200 (2007).
- [31] E. Bresson, J. Stern and M. Szydło: “Threshold Ring Signatures and Applications to Ad-hoc Groups”, CRYPTO 2002, Springer-Verlag, pp. 465–480 (2002).
- [32] M. Abe, M. Ohkubo and K. Suzuki: “Efficient Threshold Signer-Ambiguous Signatures from Variety of Keys”, IEICE Trans., **E87-A(2)**, pp. 471–479 (2004).
- [33] S. Xu and M. Yung: “Accountable Ring Signatures: A Smart Card Approach”, CARDIS, vol. 153 of IFIP, pp. 271–286 (2004).
- [34] J. Groth and M. Kohlweiss: “One-out-of-Many Proofs: Or How to Leak a Secret and Spend a Coin”, EUROCRYPT 2015, Springer-Verlag, pp. 253–280 (2015).
- [35] W. Beullens, S. Katsumata and F. Pintore: “Calamari and Falafel: Logarithmic (Linkable) Ring Signatures from Isogenies and Lattices”, ASIACRYPT 2020, Springer-Verlag, pp. 464–492 (2020).
- [36] V. Lyubashevsky, N. Nguyen and G. Seiler: “SMILE: Set Membership from

- Ideal Lattices with Applications to Ring Signatures and Confidential Transactions”, CRYPTO 2021, Springer-Verlag, pp. 611–640 (2021).
- [37] T. Yuen, M. Esgin, J. Liu, M. Au and Z. Ding: “Dualring: Generic Construction of Ring Signatures with Efficient Instantiations”, CRYPTO 2021, Springer-Verlag, pp. 251–281 (2021).
 - [38] Y. Dodis, A. Kiayias, A. Nicolosi and V. Shoup: “Anonymous Identification in Ad Hoc Groups”, EUROCRYPT 2004, Springer-Verlag, pp. 609–626 (2004).
 - [39] M. Esgin, R. Zhao, R. Steinfeld, J. Liu and D. Liu: “MatRiCT: Efficient, Scalable and Post Quantum Blockchain Confidential Transactions Protocol”, ACM CCS 2019, pp. 567–584 (2019).
 - [40] Y. Desmedt: “Society and Group Oriented Cryptography: A New Concept”, CRYPTO 1987, Springer-Verlag, pp. 120–127 (1987).
 - [41] T. ElGamal: “A Public Key Cryptosystem and Signature Scheme Based on Discrete Logarithms”, IEEE Trans. Inform. Theory, **31**, pp. 469–472 (1985).
 - [42] A. Shamir: “How to Share a Secret”, Commun. ACM, **22**, 11, pp. 612–613 (1979).
 - [43] Y. Desmedt and Y. Frankel: “Threshold Cryptosystems”, CRYPTO 1989, Springer-Verlag, pp. 307–315 (1989).
 - [44] R. Gennaro, S. Jarecki, H. Krawczyk and T. Rabin: “Robust Threshold DSS”, EUROCRYPT 1996, Springer-Verlag, pp. 354–371 (1996).
 - [45] V. Shoup: “Practical Threshold Signatures”, EUROCRYPT 2000, Springer-Verlag, pp. 207–220 (2000).
 - [46] “BBitcoin Improvement Proposal 340”. <https://github.com/bitcoin/bips/blob/master/bip-0340.mediawiki>.
 - [47] C. Komlo and I. Goldberg: “FROST: Flexible Round-Optimized Schnorr Threshold Signatures”. ePrint 2020/852.
 - [48] J. P. Aumasson, A. Hamelink and O. Shlomovits: “A Survey of ECDSA Threshold Signing”. ePrint 2020/1390.
 - [49] Y. Lindell, A. Nof and S. Ranellucci: “Fast Secure Multiparty ECDSA with Practical Distributed Key Generation and Applications to Cryptocurrency Custody”, ACM CCS 2018, pp. 1837–1854 (2018).
 - [50] R. Gennaro and S. Goldfeder: “Fast Multiparty Threshold ECDSA with Fast Trustless Setup”, ACM CCS 2018, pp. 1179–1194 (2018).

- [51] J. Doerner, Y. Kondi, E. Lee and A. Shelat: “Secure Two-Party Threshold ECDSA from ECDSA Assumptions: The multiparty Case”, IEEE S&P 2018, pp. 980–997 (2018).
- [52] I. Damgård, T. P. Jakobsen, J. B. Nielsen, J. I. Pagter and M. B. Østergaard: “Fast Threshold ECDSA with Honest Majority”, SCN 2020, Springer-Verlag, pp. 382–400 (2020).
- [53] R. Gennaro and S. Goldfeder: “One Round Threshold ECDSA with Identifiable Abort”. ePrint 2020/540.
- [54] G. Castagnos, D. Catalano, F. Laguillaumie, F. Savasta and I. Tucker: “Bandwidth-Efficient Threshold EC-DSA”, PKC 2020, Springer-Verlag, pp. 266–296 (2020).
- [55] A. Gagol, J. Kula, D. Straszak and M. Świątek: “Threshold ECDSA for Decentralized Asset Custody”. ePrint 2020/498.
- [56] R. Canetti, R. Gennaro, S. Goldfeder, N. Makriyannis and U. Peled: “UC Non-Interactive, Proactive, Threshold ECDSA with Identifiable Aborts”, ACM CCS 2020, pp. 1769–1787 (2020).
- [57] J. Groth and V. Shoup: “Design and Analysis of A Distributed ECDSA Signing Service”. ePrint 2022/506,.
- [58] A. Boldyreva: “Threshold Signatures, Multisignatures and Blind Signatures Based on the Gap-Diffie-Hellman Group Signature Scheme”, PKC 2003, Springer-Verlag, pp. 31–46 (2003).
- [59] Y. Sakemi, T. Kobayashi, T. Saito and R. S. Wahby: “Draft-irtf-cfrg-pairing-friendly-curves-10” (2021).
- [60] T. P. Pedersen: “A Threshold Cryptosystem without a Trusted Party”, EUROCRYPT 1991, Springer-Verlag, pp. 522–526 (1991).
- [61] R. Gennaro, S. Jarecki, H. Krawczyk and T. Rabin: “Secure Distributed Key Generation for Discrete-Log Based Cryptosystems”, EUROCRYPT 1999, Springer-Verlag, pp. 295–310 (1999).
- [62] M. Pettit: “Efficient Threshold-Optimal ECDSA”, CANS 2021, Springer-Verlag, pp. 116–135 (2021).

3.3 その他の高機能暗号技術

3.3.1 秘密分散

■秘密分散法の特徴 秘密分散法 (Secret Sharing Scheme, SSS) は現在、様々な暗号技術の要素技術として用いられているが、提案された当時は秘密鍵などの安全な管理を目的としていた [1, 2]。通信の秘匿性を担保するために保持しなければならない秘密鍵のような情報を保管する際に、破損や紛失を避けるためにはいくつかのコピーを作る必要がある一方で、コピーを作るとは盗難や漏洩の危険性を増大させるといった相反する問題が存在する。そこで、秘密情報を「シェア」と呼ばれる n 個の分散情報に符号化し、そのうちの任意の $k \leq n$ 個のシェアを集めることで秘密が復元できるが、 $k - 1$ 個では秘密情報が情報理論的に一切漏洩しない方式を考える。このような方式は (k, n) しきい値法と呼ばれ、 $n - k$ 個以下の破損や紛失、 $k - 1$ 個以下の盗難や漏洩に対して安全となることから、上記の問題に対する一つの解決策となる。 (k, n) しきい値法は、秘密情報へのアクセスを許されるシェア集合と、許されないシェア集合を指定可能な秘密分散法に拡張することができる。このような秘密分散法を、一般アクセス構造に対する秘密分散法と呼ぶ [3]。また、可検証秘密分散法 (Verifiable SSS、[4]) などをはじめとして様々な機能拡張がなされている。

秘密分散法は、上記のような秘密情報の分散保管を目的として提案されたが、現在ではマルチパーティ計算 (Multi-Party Computation, MPC) をはじめとした様々な暗号プロトコルの要素技術として用いられている [5]。また、様々な媒体でも実現手法が知られており、視覚暗号 [6]、量子秘密分散法 [7, 8] などが提案されている。

■秘密分散法の歴史、実装例 秘密分散法は 1979 年に Shamir [9] と Blakley [2] によって独立に提案された。どちらの論文も (k, n) しきい値法の構成手法であるが、Shamir は多項式補間 (Lagrange 補間) による方法を、Blakley は射影幾何に基づく方法を提案している。これらの手法は線形変換のランクを元にした構成法に一般化できる [10]。このような線形変換のランク関数と関連付ける研究は一般アクセス構造をもつ秘密分散法の構成法やシェアサイズの下界を求める際にも有用であり、ポリマトロイドの性質などに関連付けられて近年でも研究が進んでいる [11]。また、線形代数的なアプローチとして Monotone span program による構成法が挙げられる [12]。

文献 [10] では線形変換に基づく構成法に加えて、情報理論的な手法でシェアの

サイズの下界が導出されており、秘密情報へのアクセスが許されないシェア集合から 1 ビットも情報が漏れない完全秘匿性という性質を要求する場合、秘密情報 S を確率変数としたときの S のシャノンエントロピー（直観的にはビット長） $H(S)$ は i 番目のシェア V_i のシャノンエントロピー $H(V_i)$ に対して常に $H(V_i) \geq H(S)$ が成り立つ。Shamir の手法は任意の i に対して $H(V_i) = H(S)$ が成り立つことが容易に分かるため、Shamir の構成法がシェアサイズの観点からは最良であることになる。これらの結果を受けて、完全秘匿性を弱める代わりに、シェアのサイズを秘密情報のサイズより小さくするランプ型秘密分散法（ramp SSS）が提案された [13, 14]。ランプ型秘密分散法は Shamir の秘密分散法の一般化として構成できる。

計算量的秘密分散法は、Shamir 法と共通鍵暗号を組み合わせることで実現でき、Shamir 法に比べて大幅にシェアサイズが削減できることが知られている [15]。近年では、計算量的 (n, n) しきい値法として All-Or-Nothing Transform (AONT) が知られている。Shin らによる AONT [16] は ZenmuTech 社によって実装され、実用化されている [17]。

■秘密分散法の代表的アルゴリズム 秘密分散法は極めて多くの方式が提案されているため、最も基本的で利用頻度が多いと思われる Shamir の多項式補間によるアルゴリズム [9]、およびその拡張方式について、ISO/IEC 標準 [18] に関連するいくつかの方式を説明する。以下では全ての計算を有限体 $\mathbb{F}$ の上で行うものとする。また、自然数 n をシェアの総数として、参加者の集合を $\mathcal{P} = \{p_1, p_2, \dots, p_n\}$ とし、参加者 p_i がシェア v_i を持つとする。

Shamir 法とランプ型秘密分散法

はじめに最も基本的な Shamir の (k, n) しきい値法 [9] (Shamir 法) を説明する。秘密情報 $ss \in \mathbb{F}$ をもつディーラは、秘密情報と独立で、互いに独立かつ一様な乱数を $k - 1$ 個選び、 $r_1, r_2, \dots, r_{k-1} \in \mathbb{F}$ とする。このようにして選んだ値から、 $k - 1$ 次のランダム多項式 $f(x) := s + r_1x^1 + r_2x^2 + \dots + r_{k-1}x^{k-1}$ を作成し、参加者 P_i にシェアとして $v_i := f(i)$ を秘密通信路を介して配布する。復号にあたっては、任意の k 人の参加者 $P_{i_1}, P_{i_2}, \dots, P_{i_k}$ が各自のシェア $v_{i_1}, v_{i_2}, \dots, v_{i_k}$ をもって集まり、多項式 $f(x)$ を復元することによって秘密情報 $s = f(0)$ が復元される。多項式は Lagrange 補間によって求められるので、秘密情報は $s = f(0) = \sum_{\alpha=1}^k v_{i_\alpha} \prod_{\beta=1, \beta \neq \alpha}^k (-i_\beta)/(i_\alpha - i_\beta)$ で求められる。

完全秘匿性を達成する秘密分散法では、シェアのサイズをこれ以上短くできないことが知られている [10]。ただし、秘密情報の一部を漏らすことを許せば、シェアのサイズを小さくすることができる。このように、情報漏洩とシェアのサイズのトレードオフを実現する秘密分散法がランプ型秘密分散法である [13, 14]。ラン

プ型秘密分散法は秘密情報を $s_0, s_1, \dots, s_{L-1} \in \mathbb{F}$ として、Shamir 法の多項式 $f(\cdot)$ の代わりに $g(x) := \sum_{i=0}^{L-1} s_i x^i + \sum_{i=L}^{k-1} r_i x^i$ を用いる事で構成できる。シェアが k 個以上ある場合、Shamir 法と同様に Lagrange 補間を用いて復号が可能であり、シェアが $k-L$ 個以下なら s に関する情報が情報理論的に (完全) 秘匿される。一方で、シェアを $k-\ell$ ($1 \leq \ell \leq L-1$) 個集めた場合は秘密情報は条件付エントロピーの意味で $(1-\ell/L)H(S)$ だけ情報が漏洩する。

ただし、完全秘匿性を満たす Shamir 法では秘密情報 S はどのような確率分布に従っても良いが、ランプ型秘密分散法は $s_0, s_1, \dots, s_{L-1} \in \mathbb{F}$ を独立かつ一様ランダムに選ぶ必要があることに注意する。また、ランプ型秘密分散における情報漏洩は、秘密情報が存在しうる空間のサイズが $\mathbb{F}^L$ ではなく $\mathbb{F}$ となることを表し、秘密情報の探索空間のサイズは同じでも、秘密情報の一部が一意に定まる場合と一意に定まらない場合がある。秘密情報のどのシンボルも一意に定まらないランプ型秘密分散法は強いランプ型秘密分散 (strong ramp SSS) と呼ばれるが [14]、上に示した Shamir 法のランプ型への拡張は必ずしも強いランプ型秘密分散法になるとは限らないことが知られている [19]。

一般アクセス構造への拡張と複製型秘密分散法

しきい値型秘密分散法を用いると、一般アクセス構造に対する秘密分散法を構成することもできる。まず、参加者の集合を $\mathcal{P} = \{p_1, p_2, \dots, p_n\}$ に対する一般アクセス構造について簡単に説明する。アクセス構造とは参加者の集合 $\mathcal{A} \subseteq \mathcal{P}$ に秘密情報 $s \in \mathbb{F}$ へのアクセス権を与えるかどうかを決定するものであり、秘密情報へのアクセス権を有する有資格集合の族 $\Gamma_Q \subseteq 2^{\mathcal{P}}$ とアクセス権のない禁止集合の族 $\Gamma_F \subseteq 2^{\mathcal{P}}$ からなる。 $\Gamma_Q \cap \Gamma_F = \emptyset$ が要請される一方で、 $\Gamma_Q \cup \Gamma_F = 2^{\mathcal{P}}$ である必要はない。また、アクセス構造が自然に意味をもつために、任意の $A \in \Gamma_Q$ および任意の $A' \supseteq A$ に対して $A' \in \Gamma_Q$ であること、および、任意の $A \in \Gamma_F$ および任意の $A' \subseteq A$ に対して $A' \in \Gamma_F$ であることが必要となる。この性質を単調性と呼ぶ。

単調性をもつ任意のアクセス構造に対する秘密分散法は [3] で初めて提案されたが、ここでは [20] による、加法的秘密分散法を用いた一般アクセス構造に対する秘密分散法を説明する。

加法的秘密分散法とは、有限体における秘密鍵の加算と減算で暗号化と復号をおこなう、いわゆる使い捨て暗号 [21] の拡張であり、秘密情報 $s \in \mathbb{F}$ に対して (m, m) しきい値法を実現する方法である。具体的には、独立で一様な乱数 $r_1, r_2, \dots, r_{m-1} \in \mathbb{F}$ を用意し、 $i = 1, 2, \dots, m-1$ に対するシェアを $v_i = r_i$ とする。そして、 m 番目のシェアは $v_m = s - \sum_{i=1}^{m-1} r_i$ により計算し、 m 個のシェアとする方法である。また、秘密情報 s は m 個のシェアを全て集め、 $s = \sum_{i=1}^m v_i$

により計算できる。この加法的秘密分散法を用いた一般アクセス構造に対する秘密分散法 [20] は、任意の $A \in \Gamma_A$ に対して、 s を秘密情報とした加法的秘密分散法 $((|A|, |A|)$ しきい値法) を独立に実現し、それぞれの加法的秘密分散法で p_i に割り当てられるシェアをまとめて v_i と考えることで実現できる。

なお、[3] で提案された方式は単一の (m, m) しきい値法のシェアを複製型秘密分散法を用いて、適切に配布することで一般アクセス構造を実現している。第 3.3.2 章「MPC - 秘密分散ベース」の項で紹介する「参加者が 3 人の複製秘密分散法に基づく方法」で用いる秘密分散法はこれの特殊ケースであり、加法型の $(3, 3)$ しきい値法に対して複製型秘密分散法を適用して構成された $(2, 3)$ しきい値の秘密分散法である。

3.3.2 マルチパーティ計算-秘密分散ベース-

秘密分散ベースのマルチパーティ計算 (Multi-Party Computation, MPC) は、第 3.3.3 章に記載する Garbled Circuit ベースのマルチパーティ計算と異なり、乗算回数などに応じた通信回数が必要となる。一方で、Garbled Circuit ベースの MPC において支配的であった、共通鍵暗号による真理値表の暗号文の送受信を必要としない分、通信量の面では秘密分散ベースの MPC の方が小さくなる。この特徴から、秘密分散ベースの MPC は Garbled Circuit に基づく方法と比較して、通信量を抑えたい状況の処理に適していると考えられている。秘密分散ベースの MPC と Garbled Circuit ベースの MPC とを組み合わせる用いることもあり、特に 3 者以上の多人数の MPC において、定数回の通信による方式を実現するときなどに利用される [22, 23]。

■秘密分散ベースのマルチパーティ計算の歴史 秘密分散ベースのマルチパーティ計算の草分けとして、Goldreich らによる GMW 方式 [24] や Ben-Or らによる BGW 方式 [25] が知られている。GMW 方式は、加法的秘密分散法に基づく MPC プロトコルである。秘密情報を「シェア」と呼ばれる分散情報に分割して符号化し、シェアから秘密情報を復元することなく、秘密情報どうしの加算・乗算の計算を実現する。本章で紹介する MPC は秘密分散ベースであるため、このシェアとは、第 3.3.1 章に記載したシェアと同じものであり、元となるデータを分割したものとなる。この方式では、紛失通信と呼ばれる暗号プロトコルを利用して乗算を実現している。BGW 方式は、Shamir の秘密分散法 [1] に基づく MPC プロトコルであり、GMW 方式と同様に秘密情報のシェアを復元することなく、秘密情報どうしの加算・乗算の計算を実現する。この方式では、ローカルでの計算による乱数の偏り、多項式の次数増加を回避するため、計算結果を再分散することによって再

ランダム化・次数の削減を行って乗算を実現している。

GMW 方式と BGW 方式やその後に提案された多くの具体的な MPC プロトコルは、共通する基本的な枠組みの中で構築されている。それは、線形秘密分散法と総称される方式群に含まれる秘密分散法をベースとし、線形秘密分散法の性質を利用したローカル計算のみからなる加算プロトコルを持ち、他の参加者との通信を必要とする乗算プロトコルを備える、というものである。そのため、複雑なデータ処理を MPC で実行する際は、当該処理を論理回路/算術回路として見なした場合の AND/乗算の回数分の通信が発生し、これが処理時間全体の多くの部分を占めることになる。従って、この点に関する改善が処理全体の高速化に寄与すると考えられ、さまざまな改良がなされている。アルゴリズムの改良や実装の洗練化により、近年は実用的な処理性能が得られており、たとえば、NEC などが 7,000,000,000 ゲート/秒を達成 [26] し、また、産総研などは、データベース処理等における重要な構成要素となる秘匿状態でのシャッフルを加算・乗算のレベルまで分解せずに効率的に実現する手法を提案し、当該処理について従来の 100 倍程度の高速化に成功している [27]。

■代表的な方式 1: 参加者が 3 人の複製秘密分散法に基づく方法

効率的な方式の代表例として、参加者が 3 人の場合に向けて設計された、Araki らの方法 [26] を単純化した MPC プロトコル [28] を紹介する。以下では、各値はある整数 q を法とする剰余環の元とする。[28] の方式は、以下のような 3 つのうち任意の 2 つのシェアから復元可能な複製秘密分散法に基づいている。秘密情報 x の秘密分散は、 $x = x_1 + x_2 + x_3$ を満たす 3 つの乱数 x_1, x_2, x_3 に加法的秘密分散し、それらのうち 2 つずつの対 (x_2, x_3) 、 (x_3, x_1) 、 (x_1, x_2) をそれぞれ参加者 P_1, P_2, P_3 のシェアとすることにより実現される。復元の際は、いずれか 2 つのシェアを集め、 $x_1 + x_2 + x_3$ を計算すればよい。

加算のプロトコルは、各参加者が自身の持つ 2 つのシェアを加算することで実現される。2 つの秘密情報 x, y が秘密分散されているとき、 $z := x + y$ の秘密分散を計算する加算のプロトコルを以下に示す。各参加者 P_i ($i \in \{1, 2, 3\}$) は $z_{i+1} := x_{i+1} + y_{i+1}$ と $z_{i+2} := x_{i+2} + y_{i+2}$ を計算し、 (z_{i+1}, z_{i+2}) を z を秘密分散した P_i のシェアとする。ただし、添字の 4 は 1 と読み替えるものとし、以下も同様とする。正当性は $z_1 + z_2 + z_3 = (x_1 + y_1) + (x_2 + y_2) + (x_3 + y_3) = (x_1 + x_2 + x_3) + (y_1 + y_2 + y_3) = x + y = z$ により確認できる。このプロトコルは一切の通信を必要とせず、各参加者のローカルの計算で完結している。

乗算のプロトコルを以下に示す。入力 は加算の場合と同じとし、 $w := xy$ の秘密分散を計算するものとする。

1. 参加者間で 0 を加法的秘密分散し、各参加者 P_i ($i \in \{1, 2, 3\}$) は a_i を得る。
ここで、 a_1, a_2, a_3 は $a_1 + a_2 + a_3 = 0$ を満たす乱数である。
2. 各参加者 P_i ($i \in \{1, 2, 3\}$) は $w_{i+2} := x_{i+1}y_{i+1} + x_{i+1}y_{i+2} + x_{i+2}y_{i+1} + a_i$ を計算し、 P_{i+1} に送る。
3. (w_{i+2}, w_{i+3}) を w の秘密分散の P_i ($i \in \{1, 2, 3\}$) のシェアとする。

基本的なアイデアは、 xy を分解した $\sum_{i,j \in \{1,2,3\}} x_i y_j$ の 9 個の項のうち、参加者 P_i が自身の持つシェアから計算可能な $x_{i+1}y_{i+1} + x_{i+1}y_{i+2} + x_{i+2}y_{i+1}$ を計算結果 w の加法的秘密分散のシェアとするものである。この値は入力 x, y に関する情報を含んでいるため、ステップ 2 で隣の参加者に送る前に乱数 a_i を加算することによって見かけ上ランダムな値にしている。ステップ 1 の 0 の加法的秘密分散の作成は、各参加者が乱数を作成して次の参加者へ送り、自身の作った乱数から受け取った乱数を引いた値をシェアとすることで実現できる。正当性は、 $w_1 + w_2 + w_3 = (x_3y_3 + x_3y_1 + x_1y_3 + a_1) + (x_1y_1 + x_1y_2 + x_2y_1 + a_2) + (x_2y_2 + x_2y_3 + x_3y_2 + a_3) = (x_1 + x_2 + x_3)(y_1 + y_2 + y_3) + (a_1 + a_2 + a_3) = xy = w$ により確認できる。このプロトコルは通信を伴い、1 回の実行につき各参加者は 2 つの値を送信する。さらに、pseudorandom secret sharing [29] と呼ばれる方法を使うと、送信する乱数の一部を隣接する参加者間で共有していたシードから生成した疑似乱数で代用でき、各参加者が送信する値を 1 つにまで減らすことができる。

本方式は、NTT の秘密計算システム算師 [30] などに実装され実用化されている [31]。

■代表的な方式 2: Beaver triple を使う乗算プロトコル

参加者が 3 人の複製秘密分散法に基づく方法で述べた MPC プロトコルとは異なる特性を持つ代表的な方式として、Beaver 乗算と呼ぶ乗算プロトコルを紹介する。この方式は Beaver triple [32] と呼ばれる互いに関連した乱数の組を利用するものであり、任意の線形秘密分散法の上で構築することができる。また、事前計算を行うことにより、入力を受け取った後に必要な通信を少量にすることができるという特徴を持つ。

Beaver 乗算の基本的なアイデアは以下の通りである。乗算の実行に先立って、2 つの乱数とそれらの積をそれぞれ秘密分散したシェアの 3 つ組（これを Beaver triple と呼ぶ）を用意しておく。乗算の実行時には、まず、入力の 2 つの値に Beaver triple の 2 つの乱数をそれぞれ減算し、入力値と乱数の差分値のシェアを各参加者と通信して共有する。そして、各参加者は共有した値から差分値を復元する。差分値は乱数によりマスクされているので、復元しても入力値が参加者に

知られることはなく安全である。そして、これらの復元した差分値どうしの積を計算する。この計算は公開値どうしの計算であるから、ローカルで計算可能である。この公開値どうしの積と元々欲しかった入力との差は、Beaver triple の各値の（マスクして復元した値を定数として使った）線型結合として表現でき、線形秘密分散法の性質によりローカルで計算できる。すなわち、Beaver triple を消費することで、入力をマスクした値の復元以外はローカル計算だけで乗算が実現される。

Beaver 乗算のプロトコルを説明する。以下では、ある値 x が秘密分散された状態を $[x]$ と書くこととし、秘密分散された x と y の加減算をそれぞれ $[x] + [y]$ と $[x] - [y]$ 、秘密分散された x の定数 c による定数倍を $c[x]$ と書くことにする。Beaver triple は、 a と b を乱数、 $c := ab$ として、 $([a], [b], [c])$ とする。乗算プロトコルの入力 $[x]$ と $[y]$ とし、 $w := xy$ を満たす $[w]$ を出力とする。プロトコルは以下の通りである。

1. 各参加者は自身の持つシェアから $[\alpha] := [x] - [a]$ と $[\beta] := [y] - [b]$ を計算する。
2. 各参加者は自身の $[\alpha]$ および $[\beta]$ のシェアを他参加者に共有し、 α と β を復元する。
3. 各参加者は自身の持つシェアから $[w] := [c] + \alpha[b] + \beta[a] + \alpha\beta$ を計算して出力する。

正当性は、 $w = c + \alpha b + \beta a + \alpha\beta = ab + (x - a)b + (y - b)a + (x - a)(y - b) = ab + bx - ab + ay - ab + xy - bx - ay + ab = xy$ により確認できる。

Beaver triple は乗算プロトコルの入力とは無関係であるため、入力を受け取る前に計算して蓄えておくことができる。これにより、実際の入力を受け取った後の処理時間を小さくすることができる。Beaver triple の事前計算の方法はいくつも提案されており、例えば MPC プロトコルにより計算する方法 [33] や、信頼する第三者が作成する方法 [34] がある。

Beaver triple を使う方式は ZenmuTech 社と産総研の秘密計算ソフトウェア QueryAhead [35] などに実装され実用化されている [36]。

3.3.3 マルチパーティ計算-Garbled Circuit ベース-

■Garbled Circuit の概要 Garbled Circuit (GC) は、1986 年に Yao [37] により提案された 2 者間のマルチパーティ計算 (Multi-Party Computation, MPC) のためのフレームワークである。この方式では、2 者が協調計算したい関数を論理回路と

して表現した後、その回路の各ゲートに対応する真理値表に対し、乱数を鍵として暗号化することで、Garbled Truth Table (GTT) と呼ばれる「暗号化された真理値表」に相当する暗号文リストを作成する。この GTT に含まれる暗号文のうち、2 者の入力に対応する暗号文のみが正しく復号されるように乱数の交換を行うことで、関数値のみが復元される仕組みを実現する。

GC に基づく MPC は、第 3.3.2 章に記載の秘密分散ベースの MPC とは異なり、乗算回数などに応じた膨大な通信回数を必要とはしないが、上記の通り、処理全体をすべての入力に対する真理値表とみなしたうえで、真理値表全体の暗号文を送受信しなければならないため、通信量が非常に大きくなることが知られている。これらの特徴から、GC に基づく MPC は秘密分散ベースの MPC と比較すると遅延の大きいネットワーク環境における小規模な処理に適していると考えられる。GC と秘密分散ベースの MPC を組み合わせて用いることもあり、特に 3 者以上の多人数の MPC において、定数回の通信による方式を実現するときなどに利用される [22, 23]。

また、GC の概念は 1980 年代から知られていたものの、厳密な安全性の取り扱いについては長期にわたり十分な整備がなされていなかったが、Bellare らなどの成果 [38] により、理論的に厳密な取り扱いが可能となっている。

■代表的な方式 ここでは、GC に基づく 2 者計算の代表的な構成方法をいくつか紹介する。

以下、プロトコルに参加する 2 人の参加者を P_1 と P_2 とし、 P_1 および P_2 が持つ入力をそれぞれ $x \in \{0, 1\}^n$, $y \in \{0, 1\}^n$ として、お互いの入力を秘匿したまま関数 $\mathcal{F}(x, y)$ を計算することを目標とする。関数 $\mathcal{F}$ の論理回路による表現を C と表記する。また、 κ をセキュリティパラメータとし、 $H : \{0, 1\}^* \leftarrow \{0, 1\}^\kappa$ を暗号学的ハッシュ関数とする。

方式 1: Yao の Garbled Circuit [37]

代表的な方式の一つとして、Yao [37] による GC の構成法を紹介する。

以下の手順で回路 C に対する GC を構成することができる。

ラベル生成: 回路 C の各ワイヤ w_i および $b \in \{0, 1\}$ について、ランダムにラベル $w_i^b = (k_i^b \in \{0, 1\}^\kappa, p_i^b \in \{0, 1\})$ を生成する。ただし、 $p_i^b = 1 - p_i^{1-b}$ とする。 w_i^0 および w_i^1 は、このワイヤにおける値 0 と 1 に対応するラベルである。

GTT の生成: C に含まれる各論理ゲート G_i について、入力ワイヤに近いワイヤ

から順に以下の処理を行い、GTT を作成する。

- G_i を関数 $g_i : w_c = g_i(w_a, w_b)$ のゲートとする。入力 w_a に対応するラベルを $w_a^0 = (k_a^0, p_a^0), w_a^1 = (k_a^1, p_a^1)$ 、 w_b に対応するラベルを $w_b^0 = (k_b^0, p_b^0), w_b^1 = (k_b^1, p_b^1)$ 、出力 w_c に対応するラベルが $w_c^0 = (k_c^0, p_c^0), w_c^1 = (k_c^1, p_c^1)$ とする。
- G_i への入力 $v_a, v_b \in \{0, 1\}$ の考えられる組み合わせ 4 通り (2×2) それぞれについて、 $e_{v_a, v_b} = H(k_a^{v_a} || k_b^{v_b} || i) \oplus w_c^{g_i(v_a, v_b)}$ を計算する。
- 前ステップで計算した $\{e_{v_a, v_b}\}_{v_a, v_b \in \{0, 1\}}$ を、 $(p_a^{v_a}, p_b^{v_b})$ の順でソートする。

出力復号用テーブルの作成: 各ゲートの出力ワイヤ w_i (ラベル $w_i^0 = (k_i^0, p_i^0)$ および $w_i^1 = (k_i^1, p_i^1)$ に対応) およびワイヤが取りうる値 $v \in \{0, 1\}$ について、 $e_v = H(k_i^v || \text{"out"} || j) \oplus v$ を計算する。ただし、この計算においては H の最下位ビット (LSB) のみを用いる。得られた出力 $\{e_v\}$ を p_i^v の順でソートする。

GC に基づく MPC においては、上記の手順を P_1 が行う。この計算を行う参加者のことを Garbler と呼ぶ。

上記の手順により生成した出力復号用テーブルを含む GC を、 P_1 から P_2 へ送信する。同時に、各入力ワイヤについて、 P_1 自身の入力に対応するラベルも P_2 へ送信する。

GC の “復号” には、 P_2 の入力に対応するラベルを P_1 から P_2 の入力を知ることなく P_2 へ送信する必要がある。このような機能は、紛失通信 (Oblivious Transfer, OT) [39] によって実現することができる。各 P_2 の入力ワイヤ w_b について、 P_1 は 0, 1 それぞれの入力に対するラベル w_b^0, w_b^1 を入力し、 P_2 は選択ビット c を入力することで、 P_1 から P_2 へ w_b^c のみを送信することができる。

以降は P_2 が受け取った入力を基に GC の評価を行う。この手順を行う参加者を Evaluator と呼ぶ。各ゲートに対応する GTT $T = (e_{0,0}, e_{0,1}, e_{1,0}, e_{1,1})$ および入力ラベル $w_a = (k_a, p_a), w_b = (k_b, p_b)$ を用いて、 P_2 は $w_c = H(k_a || k_b || i) \oplus e_{p_a, p_b}$ を計算する。

すべてのゲートを評価し終わったら、最終的な出力を復号するために、第 2 キーに “out” を用いて、計算の平文結果となる最終出力を得ることができる。 P_2 は得られた出力を P_1 に返却することで、両者が結果を得ることができる。

方式 2: Half-Gates [40]

GC においては、GTT のサイズを削減するためにいくつかのテクニックが知られている。Yao の方式では 2 入力 1 出力のゲート 1 つあたりに 4 つの暗号文を必要

としたが、XOR ゲートの暗号文数を 0 に削減する Free-XOR と呼ばれるテクニック [41] や、一般のゲートにおける暗号文数を 3 あるいは 2 へ削減する Garbled Row Reduction (GRR) テクニック [42, 43] などが知られる。

これらのテクニックを組み合わせた方式で効率が良い方式の一つとして、Zahur らによって提案された Half-gates [40] と呼ばれる方式を説明する。この方式では、XOR ゲートの GTT を必要とせず、AND ゲートの GTT を暗号文 2 つで構成することが可能である。AND ゲートと XOR ゲートの組み合わせで、任意の回路を表現可能であることに注意する。

Half-gates のアイデアのポイントは、AND ゲートを 2 つの “half gate” の XOR として表現することである。このそれぞれの half gate は、入力の一つが参加者のいずれかに既知の AND ゲートであり、暗号文 2 つで GTT を構成することができる。この暗号文 2 つの GTT を GRR テクニックによって暗号文 1 つに圧縮し、2 つの half gate を接続する XOR を Free-XOR テクニックによって削減することで暗号文 2 つによる AND ゲートの GC を達成している。

2 つの half gate はそれぞれ、“generator half gate” と “evaluator half gate” と呼ばれる。以下でそれぞれについて説明する。

• Generator half gate

入力ワイヤを a と b 、出力ワイヤを c とする AND ゲートを考える。Generator half gate は $v_c = v_a \wedge v_b$ のような回路で、 v_a が generator (P_1) に対して何らかの形で既知であり、evaluator (P_2) はいずれの入力も未知の回路である。このとき、 $v_a = 0$ であれば $v_c = 0$ であり、 $v_a = 1$ であれば $v_c = v_b$ であることに注意する。

いま、 k_a^0, k_b^0, k_c^0 をワイヤ a, b, c それぞれの入力 0 に対応するラベルとする。このとき、generator half gate の GTT は以下のように表される。

$$\begin{aligned} H(k_b^0) \oplus k_c^0 \\ H(k_b^1) \oplus k_c^0 \oplus v_a \cdot R \end{aligned}$$

ただし、 $k_b^1 = k_b^0 \oplus R$ であり、 R はランダムなグローバルオフセットである。Free-XOR を用いる際はこのように、0 と 1 それぞれに対応する 2 つのラベルを $(X, X \oplus R)$ のように差分を固定した形で設定する。

Evaluator (P_2) はこの half gate を評価するために、自身の入力 b に対応するラベル (k_b^0 あるいは k_b^1) を用いてそのハッシュ値を計算する。もしワイヤ b の値が 0 の場合、evaluator は k_b^0 を持ち、 $H(k_b^0)$ を計算して k_c^0 を得ることができる。一方、もし b の値が 1 の場合は、 $k_b^1 = k_b^0 \oplus R$ によって $H(k_b^1)$ を計算して $k_c^0 \oplus v_a \cdot R$ を得ることができる。ちなみに、 $v_a = 0$ のとき k_c^0 となり、 $v_a = 1$ のとき $k_c^0 \oplus R$ となる。

- Evaluator half gate

Evaluator half gate は、 $v_c = v_a \wedge v_b$ のような形で、 v_a が evaluator (P_2) にとって既知であり、generator (P_1) はいずれの入力も未知の回路である。このとき、evaluator は自分の知るワイヤ a の入力に応じて異なる振る舞いを行うことができる。Evaluator half gate の GTT は以下のように実現される。

$$\begin{aligned} H(k_a^0) \oplus k_c^0 \\ H(k_a^1) \oplus k_c^0 \oplus k_b^0 \end{aligned}$$

ただし、generator half gate と同様、 $k_a^1 = k_a^0 \oplus R$ であり、 R はグローバルなオフセットである。ワイヤ a の値が 0 の場合、 $H(k_a^0)$ を計算し、 k_c^0 を得ることができる。一方、 a の値が 1 の場合は、 $k_c^0 \oplus k_b^0$ を得ることができる。最終的に、これを b に対応するラベルと XOR を取ることで、 b の値が 0 の場合、すなわち evaluator が k_b^0 を持っている場合は k_c^0 を、 b の値が 1 の場合、すなわち、evaluator が k_b^1 を持っている場合は $k_c^1 = k_c^0 \oplus R$ を得ることができる。

- Generator half gate/Evaluator half gate の結合

上記の 2 種類の half gate を用いて $v_c = v_a \wedge v_b$ のゲートを GC で評価するために、generator は GC の生成時にランダムなビット r を生成し、これを用いて AND ゲートを以下のように変形する。

$$\begin{aligned} v_c &= v_a \wedge (r \oplus r \oplus b) \\ &= (v_a \wedge r) \oplus (v_a \wedge (r \oplus v_b)) \end{aligned}$$

$(v_a \wedge r)$ は generator half gate を用いて garbling することができ、 $(v_a \wedge (r \oplus v_b))$ は point-and-permute [22] を用いると $r \oplus v_b$ を evaluator に伝えることができるため、evaluator half gate を用いて garbling することができる。XOR ゲートについては上記の通り、Free-XOR テクニックを用いると GTT が不要であるため、GRR で圧縮した generator half gate の GTT と evaluator half gate の GTT の組が、所望の AND ゲート $v_a \wedge v_b$ に対する GTT となる。

3.3.4 ゼロ知識証明

■**ゼロ知識証明の特徴** ゼロ知識証明とは、ある命題 X に対して、証明者と検証者の間で情報をやりとりし、証明者が検証者に命題 X が正しいことを確信させ、かつ命題が正しいこと以外の情報を検証者に与えないプロトコルである。次に述べる完全性、健全性、ゼロ知識性の性質をもつ。

完全性： 証明者の命題が真ならば、検証者は真であることが必ずわかること。

健全性： 証明者の命題が偽ならば、検証者はかなり高い確率で、偽であることを見抜けること。

ゼロ知識性： あらゆる場合で検証者が証明者から何らかの知識を得ようとしても、証明者の命題が真であること以上の知識は得られないこと。

さらに、ゼロ知識証明の中でも zk-SNARKs 方式 (Zero-Knowledge Succinct Non-Interactive Argument of Knowledge) は、命題が正しいことの証拠の保有を証明するもので、名前に由来する以下の特徴を持つ。

簡潔 *Succinct*： 知識証明のサイズが命題のサイズに依存せず一定で、十分に短い

非対話 *NonInteractive*： 検証者は証明者からの一度の知識証明の送付で検証可能

アーギュメント *Argument*： 検証者は計算能力に限りある (多項式時間計算力) 証明者からは守られる

知識 *Knowledge*： 証明者は証拠なしでは知識証明の生成は不可能

ゼロ知識証明で扱える命題は、その証拠を使って多項式時間アルゴリズムで真偽判定できる「クラス NP 問題」であり、関数の解や充足可能問題なども含む幅広い。

■**ゼロ知識証明の歴史** ゼロ知識証明の歴史は古く、1985 年に Goldwasser らにより、証明者と検証者間で対話を繰り返し、知識を証明するプロトコルとして定式化された [44]。この対話型の改良として、信頼できる第三者機関が、共通参照情報 (Common Reference String, *CRS*) を事前に生成し、配布することで、非対話型が達成される [45]。また健全性について、離散対数問題をベースとしたシュノアプロトコル [46] は、証明者の計算能力を限定することで健全性を保証する方式で、計算困難な数学的問題を根拠とする方式の原型といえる。

クラス NP 問題に、一方向性関数の存在を仮定すると、非対話型ゼロ知識証明が適用できることは保証されていたが、実用的な手法は 2013 年の Gennaro らによる zk-SNARK である [47]。これを一般化した zk-SNARKs 方式の中でも Pinocchio [48] は、暗号資産の取引システム上で実用化されている。また Pinocchio の効率化が Groth により定式されている [49]。さらに、zk-STARKs [50] と BulletProofs [51] は、証明サイズや計算量は大きくなるが、事前準備を必要としないことを特徴とする。

■代表的なゼロ知識証明方式 Pinocchio アルゴリズムの概要を説明する。はじめに、命題「多項式 $\mathcal{F}$ の解の存在」をゼロ知識証明する場合を例に、全体の流れを示す。

1. $\mathcal{F}$ を加算と積算演算ゲートを持つ算術回路に変換する。
2. 具体的な手続きに従って、回路を 2 次算術プログラム (QAP) で表現する。
ここで QAP は 3 つの多項式の集合 V, W, Y と目的多項式 $t(x)$ で構成される。回路を満たす割り当て $\{c_i\}$ がある時、 V, W, Y と $\{c_i\}$ から多項式 $p(x)$ が定まり、後述の QAP 定理 [47] より「 $\mathcal{F}$ の解の存在」と「 $h(x)t(x) = p(x)$ を満たす多項式 $h(x)$ の存在」が同値になる。
3. 証明者は、証拠から $h(x)$ を計算し、 $h(x)$ を暗号化した知識証明 π を作り、検証者に渡す。
4. 検証者は、知識証明 π から、 $h(x)t(x) = p(x)$ の成立を、Pairing 写像を用いて検証する。

以上のプロセスを、非対話型にするために CRS と呼ばれる乱数の共有が必要である。第 2 項にある QAP 定理を以下に示す。

QAP 定理 [47]: $\mathcal{F}$ を n 個の元を入力し n' 個の元を出力する関数とし、 $N = n + n'$ とする。この時、3 つの多項式集合 $V = \{v_k(x)\}, W = \{w_k(x)\}, Y = \{y_k(x)\}_{k=1, \dots, m}$ と目的多項式 $t(x)$ が存在し (これを $\mathcal{F}$ の QAP と呼ぶ)、次が成立する。入出力値 $(c_1, \dots, c_N)$ が $\mathcal{F}$ から得られる回路の正しい割り当て (値) とすると、係数 $(c_{N+1}, \dots, c_m)$ が存在し、 $t(x)$ が $p(x)$ を割り切る。この時、係数 $(c_{N+1}, \dots, c_m)$ を証拠と呼ぶ。ここで $p(x)$ と $t(x)$ は以下の式となる。

$$p(x) = \left(\sum_{k=1}^m c_k v_k(x) \right) \left(\sum_{k=1}^m c_k w_k(x) \right) - \sum_{k=1}^m c_k y_k(x),$$

$$t(x) = \prod_{g: \text{積算演算ゲート}} (x - r_g)$$

具体例として、 $\mathcal{F}(C_1, C_2, C_3, C_4) = (C_1 \times C_2) \times (C_3 + C_4) = C_6$ を当てはめると、 $\mathcal{F}$ の QAP は目的多項式 $t(x) = (x - r_5)(x - r_6)$ と $V = \{v_k(x)\}, W = \{w_k(x)\}, Y = \{y_k(x)\}_{k=1, \dots, 6}$ で構成される。 V の初項は $v_1(x) = (x - r_5)(x - r_6 + 1)/(r_6 - r_5)$ であり、 $i \geq 2$ の $v_i(x)$ と W, Y は順次計算できるが、残りの項の記載は省略する。そして、 $\mathcal{F}$ を満たす解 $x = (c_1, c_2, c_3, c_4, c_6)$ と途中ゲートの入出力制約をみたす c_5 を使用して、 $p(x) = \{c_1 v_1(x) + c_2 v_2(x) + c_3 v_3(x)\} \{c_4 w_4(x) + c_5 w_5(x)\} - \{c_5 y_5(x) + c_6 y_6(x)\}$ を計算する。 $p(x)$ を展開すると、「 $(c_1 \times c_2) \times (c_3 + c_4) = c_6$

の成立」と「 $t(x)$ が $p(x)$ を割り切る」が同値であることが分かる。

定式化のため写像を 2 つ用意する。1 つ目は巡回群 G とその生成元 g による、暗号化写像 $E(s) := g^s$ であり、単射かつ $E(s+t) = E(s)E(t)$ を満たす。値 g^s と g からは s が計算困難な離散対数困難性を持つ。2 つ目は G 上の Pairing 写像 e で、双線形性 $e(g^a, g^b) = e(g, g)^{ab}$ を持つ。Pinocchio は (CRSGen, Proof, Verify) の 3 つのアルゴリズムからなる。

CRSGen($1^\lambda, \mathcal{F}$): 信頼できる第三者により CRS を生成するアルゴリズム。

証明したい命題 $\mathcal{F}$ を、算術回路変換を通して QAP 表現する。QAP がサイズ m 、次数 d 、 $V = \{v_k(x)\}, W = \{w_k(x)\}, Y = \{y_k(x)\}_{k=1, \dots, m}, t(x)$ で構成される時、 $[N] = \{1, \dots, N\}, I_{mid} = \{N, \dots, m\}, [d] = \{1, \dots, d\}$ と置く。 $s, \alpha, \beta_v, \beta_w, \beta_y, \gamma$ をランダムに選び、 $CRS = (EK, VK)$ を生成する。証明者に EK を、検証者に VK を送付する。

$$EK = \{\{E(s^i), E(\alpha s^i)\}_{i \in [d]}, \{E(v_k(s)), E(\alpha v_k(s)), E(\beta_v v_k(s))\}_{k \in I_{mid}}, \\ \{E(w_k(s)), E(y_k(s)), E(\alpha w_k(s)), E(\alpha y_k(s)), E(\beta_w w_k(s)), \\ E(\beta_y y_k(s))\}_{k \in [N]}\}$$

$$VK = \{E(1), E(\alpha), E(\gamma), E(\beta_v \gamma), E(\beta_w \gamma), E(\beta_y \gamma), E(t(s)), E(v_k(s))_{k \in [N]}\}$$

Proof(EK, u): 証明者により知識証明 π を生成するアルゴリズム。

証明者は、入力 u から $y = \mathcal{F}(u)$ を計算し、回路が満たす $\{c_k\}_{k \in [m]}$ を得る。これから

$$p(x) = \left(\sum_{k=1}^m c_k v_k(x)\right) \left(\sum_{k=1}^m c_k w_k(x)\right) - \sum_{k=1}^m c_k y_k(x) \\ h(x) = p(x)/t(x)$$

と多項式 $h(x)$ を得る。次に、この $h(x)$ の暗号化を行う。 $h(x)$ の係数 h_i から

$$E(h(s)) = \prod_{i \in [d]} E(s^i)^{h_i}$$

を計算する。さらに、秘匿したい入出力値と証拠 $\{c_k\}_{k \in I_{mid}}$ も使い、 $E(v_{mid}(s)) = \prod_{k \in I_{mid}} E(v_k(s))^{c_k}$ や $E(w(s))$ など同様の計算を行い、合わせて、知識証明 π を構成する。

$$\pi = \{E(h(s)), E(v_{mid}(s)), E(w(s)), E(y(s)), E(\alpha h(s)), \\ E(\alpha v_{mid}(s)), E(\alpha w(s)), E(\alpha y(s)) E(\beta_v v(s)) E(\beta_w w(s)) E(\beta_y y(s))\}$$

π に、 y と u の中で秘匿しない入出力値に対応する $\{c_k\}$ を加えて、検証者に送る。

しかし、上記の定式化では、 π のゼロ知識性が満たされていない。そのため、一様乱数の組と区別できないように、 $v'_{mid}(x) = v_{mid}(x) + \delta t(x)$ と、暗号化前の各多項式に乱数 δ を加えてゼロ知識性を達成する。これに合わせて $p(x)$ と $h(x)$ も変更される。

Verify(VK, u, y, π): 検証者により証明者の知識 π を検証するアルゴリズム。

次の 2 種類のチェックのすべてが通った時に、検証成立となる。

1. 等式 $p(s) \stackrel{?}{=} h(s)t(s)$ の成立チェック

VK と回路の秘匿しない入出力値 $\{c_k\}$ を使い、 $E(v_{i/o}(s)) = \prod_{k \in [N]} E(v_k(s))^{c_k}$ を計算する。そして、

$$e(E(v_{i/o}(s))E(v_{mid}(s)), E(w(s))) / e(E(y(s), E(1))) \stackrel{?}{=} e(E(h(s)), E(t(s)))$$

の両辺を比較し一致する時、 e の双線形性より $(v_{i/o}(s) + v_{mid}(s))w(s) - y(s) = h(s)t(s)$ が成立する。

2. 線形結合チェック

$$\begin{aligned} e(E(\alpha h(s)), E(1)) &\stackrel{?}{=} e(E(h(s)), E(\alpha)), \\ e(E(\beta_v v(s)), E(\gamma)) &\stackrel{?}{=} e(E(v(s)), E(\beta_v \gamma)) \end{aligned}$$

を含む、 α に関する 8 ペアと β に関する 3 ペアの Pairing 写像を用いたチェックを行う。

■ゼロ知識証明の安全性など アルゴリズムを支える根拠と安全性を述べる。Proof の知識証明 π は回路のサイズに関わらず巡回群 G の 8 つの元で構成され、サイズは一定となる。さらに、多項式自身でなく、ランダム s での評価値 $h(s)$ が知識証明に使用されている。これは Schwartz-Zipple 補題により、異なる多項式をランダムな点で評価すると高い確率で異なる値となることが示されており、このようなランダム点での評価値を用いることができる。さらに、多項式の秘匿評価手法により CRS の $E(s^i)$ を使うことで s 自身を知らずとも $E(h(s))$ が計算可能である。Verify において、線形結合チェックが加えられている。これは、証明者は $h(x)$ や $v_{mid}(x)$ を、証拠（の係数）を持たずとも、等式が成立する任意の多項式として作ることができる。この不正な攻撃に対応するために、検証者は $h(s)$ が $\{s^i\}_{i \in [d]}$ の線形結合から作られていることの確認が必要である。Knowledge of Coefficient (KC) に基づくと、 $e(E(\alpha h(s)), E(1)) = e(E(h(s)), E(\alpha))$ が成立する時に、この

線形結合性が確認できる。そして、検証は一度の知識証明の送付で達成しており、非対話型である。

3.3.5 Oblivious Random Access Machine (ORAM)

■ORAM の特徴 外部サーバにデータを預ける際、暗号化していればデータの内容は漏洩しないが、あるデータへのアクセスの偏りが発生した場合、サーバはそのアクセスの偏りから該当データの情報を推測できてしまう可能性がある。ORAM では、暗号化データに対して読み込み書き込みを行うたびに、格納位置をシャッフルするとともに再暗号化することで各アクセス間の関係を秘匿し、サーバに対してアクセスパターンを秘匿できる。クライアントは格納位置に関する情報を保持することによって、目的の暗号化データの格納位置を知ることができる。

同様にサーバに対するプライバシー保護を目的とする高機能暗号技術である第 3.3.6 章に記載の Private Information Retrieval (PIR) と第 3.3.7 章に記載の検索可能暗号との違いは以下の通りである。

- サーバへのクエリ内容：検索可能暗号では格納したデータがあるキーワードを含んでいるかどうかをクエリするのに対し、ORAM と PIR ではクライアントがアクセスしたいデータそのものについてクエリする。
- 格納したデータの暗号文：検索可能暗号と PIR では基本的にサーバに格納されたデータやその暗号文は（データの更新処理などが行われない限り）クエリによって変わることが無いのに対し、ORAM では毎回のクエリごとにデータの更新と暗号文の再暗号化、格納位置のシャッフルが行われる。
- 秘匿できる情報：検索可能暗号では検索に該当するデータの頻度などの情報は秘匿しないのに対し、ORAM と PIR では頻度を含めたアクセスパターンを秘匿する。
- クライアントとサーバの処理：検索可能暗号と PIR ではクエリに応じてサーバが暗号化データベースに対して検索処理を行うのに対し、ORAM では暗号化データベースの一部をサーバから受け取りクライアントが検索処理（およびにシャッフルと再暗号化）を行う。

このように、検索可能暗号はアクセスパターンの秘匿までは必要のない状況、PIR は通信量とサーバの計算量が増えてもクライアントの計算量を抑えたい状況、ORAM はある程度クライアントは計算量を要するが通信量を抑えたい状況で使い分けることが望ましい。

■ORAM の歴史と方式の比較 ORAM の概念は、Goldreich [52] によって初めて提案された。元々はソフトウェア保護の文脈で CPU によるメモリアクセスパターンの秘匿を目的として考案されたが、その後、CPU をクライアント、メモリをサーバに置き換えることにより、暗号化データベースのアクセスパターン秘匿への適用が主な応用先となった。Goldreich と Ostrovsky による初期の方式 [52, 53, 54] では、データブロック数 N に対してアクセスごとの償却通信計算量は $\mathcal{O}(\log^4 N)$ であったが、最悪通信計算量は $\Omega(\log N)$ であった。

その後、Shi ら [55] や Kushilevitz ら [56] によって、木構造を用いてデータを管理することによって最悪通信計算量がそれぞれ $\mathcal{O}(\log^3 N / \log \log N)$ と $\mathcal{O}(\log^4 N)$ となる方式が提案された。通信計算量の観点で効率の良い ORAM 方式として、Stefanov ら [57] による Path ORAM が知られており、データブロック数 N に対して通信計算量は $\mathcal{O}(\log^3 N)$ でよい。ただし、Path ORAM では、クライアントが position map と呼ばれるテーブルとスタッシュと呼ばれるローカルなメモリに情報を格納するため、クライアントの記憶容量は N に依存したサイズが必要である。

近年はクライアントの記憶容量や通信計算量を漸近的に減らす研究が進んでいる。Patel ら [58] は $\mathcal{O}(\log N)$ ビットのクライアント記憶容量かつ $\mathcal{O}(\log^2 N \log \log N)$ の償却通信計算量の方式を提案した。Asharov ら [59] はこれを改良し、 $\mathcal{O}(\log^2 N)$ の償却通信計算量の方式を提案した。さらに、Asharov ら [60] により $\mathcal{O}(\log N)$ ビットのクライアント記憶容量かつ $\mathcal{O}(\log^2 N)$ の最悪通信計算量の方式が提案された。この方式は漸近的には効率的であるものの、オーダー表記に隠れた定数倍が大きいため実用は現実的ではない。

ORAM の拡張として、サーバを複数台に分けてデータを格納することによって、単一サーバの ORAM より効率が良い分散 ORAM [61] が知られている。また、通常の ORAM ではクライアントは単一だが、複数のクライアントを含むことができる並行 ORAM [62] も知られている。

■代表的な方式のアルゴリズム 具体的な ORAM の実現方法として、木構造を用いた Path ORAM [57] を紹介する。

- サーバのセットアップ： 高さ L の二分木を生成する。各ノードにそれぞれ Z 個のブロックを割り当て、暗号化データを格納する。もしあるノードに入るブロックが Z 個未満になる場合は、真のブロックとは識別できないダミーのブロックを格納することで常に Z 個のブロックで満たされるようにする。
- クライアントのセットアップ： それぞれのブロックと葉ノードを紐づけた

position map と呼ばれるテーブルを生成する。position map は、ブロックが根ノードから紐づけられた葉ノードの経路上のいずれかにあることを保証し、あるブロック a が左から x 番目の葉ノードと紐づけられているならば、 a は根ノードから x 番目の葉ノードへの経路上のいずれかにあることになる。初期状態においてブロックは全て空の状態、各ブロックはそれぞれランダムに選ばれた葉ノードと紐付けられて、その葉ノードの経路上に配置される。また、いくつかのブロックを保存できるスタッシュと呼ばれるメモリを用意する。

- アクセス：

1. 処理 $op \in \{read, write\}$ 、ブロック a 、データ d に対して、クライアントは position map からブロック a に対応した葉ノード x を選択し、 x をサーバに送信する。その後 position map の a に対応した葉ノードをランダムな葉ノード x' に上書きする。
2. サーバは受け取った x から根ノードへの経路上にあるブロックを全てクライアントに送る。
3. クライアントはサーバから受け取ったブロックをスタッシュに保存し、スタッシュにある暗号化データを全て復号する。 a を用い参照したいデータを探し出し、 $op = write$ ならば内容を d に更新する。
4. クライアントは全てのデータを再暗号化し、position map と x の最小共通祖先ノードに格納する。参照した a は、 x と x' の最小共通祖先ノードに格納される。この時ノードに空きが出来た場合、ダミーのブロックを格納する。入るべきノードがすべて埋まってしまいスタッシュに残ったブロックは、そのままスタッシュに一時保存する。

3.3.6 Private Information Retrieval (PIR)

■PIR の特徴および他の関連技術との差異 クライアントがサーバ上のデータを検索する際、クライアントは検索のためのクエリをサーバに送信し、そのレスポンスとして、クライアントはサーバから検索結果を受け取る。これら一連の作業において検索履歴や検索結果、サーバに保存されたデータ自体の秘匿性をどのように考えるかによって様々な方式が提案されている。秘匿情報検索 (Private Information Retrieval, PIR) は、サーバ上のデータは、暗号化されず平文のまま保存されている状況で、クライアントがどのデータにアクセスしたか、サーバに秘匿することを目的とする。サーバに預けたデータをクエリする際に、そのアクセス情報を秘

匿する暗号技術としては、PIR の他に第 3.3.5 章に記載された Oblivious Random Access Machine (ORAM) や第 3.3.7 章に記載された検索可能暗号がある。これらの暗号技術と PIR の具体的な相違点は以下の通りである。

- サーバが保持する情報：PIR ではデータを暗号化しない状態で保持する一方、ORAM や検索可能暗号ではデータを暗号化した状態で保持する。言い換えれば、PIR では元々サーバが管理するデータを、ORAM 及び検索可能暗号ではクライアントが管理するデータを扱うことを想定している。
- 格納した暗号化データの取り扱い：PIR では基本的にサーバに格納されたデータやその暗号文はクエリによって変わることが無いのに対し、ORAM や検索可能暗号では毎回のクエリごとにデータの更新と暗号文の再暗号化等が行われる。特に ORAM では格納位置のシャッフルも行われる。
- サーバへのクエリ内容：ORAM と PIR ではクライアントがアクセスしたいデータそのものについてクエリするが、検索可能暗号では格納したデータがあるキーワードを含んでいるかどうかをクエリする。
- サーバに対して秘匿できる情報：いずれの技術もクエリ内容を秘匿する。特に、クエリに対してどのデータが返されたかに関する情報を秘匿する。ORAM と PIR では頻度を含めた全ての情報を秘匿するが、検索可能暗号では検索に該当するデータの頻度などの一部の情報が漏洩することを許容する。

PIR の具体的な設定は以下の通りである。クライアント U と t 個のサーバ $DB_1, DB_2, \dots, DB_t$ がある。各サーバは同じデータ $x = (x_1, x_2, \dots, x_n) \in \{0, 1\}^n$ をもっており、結託はしないものとする。 U が x_i を検索するために各サーバにクエリを送信し、サーバがレスポンスを返す。必要に応じてこれを繰り返すことでクライアントは x_i を得る。安全性要件として、クライアントが送信するクエリは、クライアントが検索しようとしているデータのインデックス i をサーバに漏らさないことを要請する。この安全性要件に対して、PIR では、情報理論的安全性、計算量的安全性を満たすプロトコルがそれぞれ提案されている。

このような設定のもとで自明なプロトコルは、 $t = 1$ として、クライアントはクエリを生成せず、サーバが全てのデータ $x_1, x_2, \dots, x_n$ をサーバに送ることである。従って、PIR を考えるとき、この自明な方式よりは効率的な手法であることが求められる。

■PIR の歴史 PIR は Chor らによって最初に提案された [63, 64]。ここで提案された方式は情報理論的安全性を満たす。彼らは PIR を情報理論的に達成するために

は（自明な方式を除いて）サーバ 1 台では不可能であることを示し、結託しない 2 台のサーバが同じデータのコピーをもっていれば PIR が実現できることを示した。この結果から、計算量的安全性のもとで、サーバの数を 1 台まで減らせるかが重要な課題となったが、これは Kushilevitz ら [65] によって肯定的に解決された。この方式でのサーバの計算時間は $\Omega(n)$ であり、平方剰余を用いていた。近年の方式では完全準同形暗号を用いて、さらに前処理を含めた複数ラウンドのプロトコルで劣線形時間で実行できるプロトコルが知られている [66, 67]。また符号に基づく PIR も Holzbaur らによって提案されている [68, 69]。

情報理論的な観点からは、 t 個のサーバからダウンロードするデータ 1 ビットに対して、検索可能なビット数（通信レート）の最大値（通信容量）に興味がある。例えば、PIR の特徴および他の関連技術との差異の項で示した自明な方式のレートは $1/n$ である。Sun らはこの最大値が $1/\sum_{u=0}^{k-1} n^{-u}$ であることを示し、実際にこの容量を与えるプロトコルを示した [70]。この研究に端を発して、情報理論的な PIR の研究は符号理論などの情報理論的な技術と関連して活発になってきた。例えば、 t 台のサーバに同一のコピーを置かずに、サーバに置くデータサイズの削減を目指した Coded PIR [71] は、Katz らによる Locally decodable code [72] を用いて構成される。また、複数のサーバからの応答のうち、いくつかが破損したり改竄されている場合にも正しく情報が得られるロバスト PIR に関する研究も行われている。ロバスト PIR は Beimel と Stahl によって提案され [73]、Belekamp–Welch アルゴリズムによる効率化が Kurosawa によって提案されている [74]。

■代表的方式 本節では、情報理論的に安全な方式として [64] を、計算量的に安全な方式として単一サーバの PIR として初の方式である [65] を紹介する。

情報理論的に安全な 2 サーバ PIR [64]

[64] では、結託しない 2 台のサーバ DB_1, DB_2 を準備し、それぞれが n ビットのデータ $x := (x_1, x_2, \dots, x_n) \in \{0, 1\}^n$ を保持しているものとする。ここでクライアントが x_i の値を知りたいとすると、以下のようなプロトコルを実行する。

自然数 n に対して、 $[n] := \{1, 2, \dots, n\}$ と定める。

- 1) $2^{[n]}$ 上の一様分布から集合 $S \subseteq [n]$ を選ぶ。
- 2) $S_1 := S$ として、 $s \in S$ であるなら、 $S_2 := S \setminus \{i\}$ 、 $s \notin S$ であるなら $S_2 := S \cup \{i\}$ として集合 S_1, S_2 をそれぞれ DB_1, DB_2 に送る。
- 3) $i = 1, 2$ に対して、集合 S_i を受け取った DB_i は $r_i = \bigoplus_{j \in S_i} x_j$ をクライアントに返送する。ここで $\oplus$ は XOR を表す。

4) クライアントは $r_1 \oplus r_2$ として x_i を求める。

プロトコルの健全性は方式からほぼ明らかである。また、集合 S が $2^{[n]}$ から一様分布に従って選ばれているため、 i が S に属す確率は $1/2$ である。従って、 i が S_1 に属す確率も $1/2$ であり、 S_2 に属する確率も $1/2$ となる。このことから、 i の情報は DB_1, DB_2 のどちらにも情報理論的な意味で漏洩しない。

計算量的に安全な単一サーバ PIR [65]

情報理論的な設定の下で、サーバの数を 1 つにすることができないが [64]、計算量的安全性のもとではサーバを 1 台にすることができる。以下に、Kushilevitz ら [65] によって初めて提案された単一サーバによる PIR プロトコルを示す。

本方式は一人のクライアントとサーバ DB のみで構成され、サーバには s 行 t 列の二値行列 $x \in \{0, 1\}^{s \times t}$ が格納されているものとする。クライアントが行列 M の第 a, b 成分、 $M_{a,b}$ ($1 \leq a \leq s, 1 \leq b \leq t$) をの値を知りたいとするととき、以下のようなプロトコルを実行する。

- 1) 自然数 k に対して、クライアント U は $k/2$ ビット素数をランダムに 2 つ選び、その積を N とする。
- 2) クライアント U は t 個の自然数 $y_1, y_2, \dots, y_t \in \{y \in \mathbb{Z}_N^* \mid (\frac{y}{N}) = 1\}$ を、 y_b は平方非剰余、それ以外は平方剰余であるように一様ランダムに選ぶ。ここで $(\cdot)$ は Jacobi 記号である。
- 3) 行列 M の第 i, j 成分 $M_{i,j} \in \{0, 1\}$, $1 \leq i \leq s, 1 \leq j \leq t$ に対して、サーバ DB は $M_{i,j} = 0$ のときに $w_{i,j} := y_j^2$, $M_{i,j} = 1$ のときに $w_{i,j} := y_j$ として $W := (w_{i,j})_{(i,j) \in [s] \times [t]}$ を作成する。DB はさらに、 W から $z_i := \prod_{j=1}^t w_{i,j}$, $i = 1, 2, \dots, s$ を計算する。
- 4) サーバは $z_1, z_2, \dots, z_t$ をクライアントに送る。
- 5) クライアントは z_a を取り出し、 z_a が N を法として平方剰余なら $M_{a,b} = 0$ 、平方非剰余であるなら $M_{a,b} = 1$ とする。

構成法から、 $M_{a,b} = 1$ のときに限って z_a は N を法として平方剰余となる。クライアントは N の 2 つの素因数を用いて、任意の $y \in \mathbb{Z}_N^*$ が N を法とした平方 (非) 剰余であるかを多項式時間で判定できるため、 $M_{a,b}$ の値が計算できる。一方で、サーバは N の 2 つの素因数を知らないため、平方剰余仮定により y_j や z_i が平方剰余であるか否かを判定できない。このことから、直観的にはサーバが a, b を特定出来ないことがわかる。

3.3.7 検索可能暗号

検索可能暗号とは、サーバに保存されている（暗号化）文書に対して、クライアントが検索したいキーワード及び暗号化文書の内容を秘匿しながらも、当該キーワードを含む（暗号化）文書のみを効率的に検索できる高機能暗号技術である。特に関連する高機能暗号技術として、第 3.3.5 章に記載された Oblivious Random Access Machine (ORAM) や第 3.3.6 章に記載された Private Information Retrieval (PIR) がある。以下に各技術の違いをまとめる。

- サーバが保持する情報：PIR ではデータを暗号化しない状態で保持する一方、ORAM や検索可能暗号ではデータを暗号化した状態で保持する。言い換えれば、PIR では元々サーバが管理するデータを、ORAM 及び検索可能暗号ではクライアントが管理するデータを扱うことを想定している。
- 格納した暗号化データの取り扱い：PIR では基本的にサーバに格納されたデータやその暗号文はクエリによって変わることが無いのに対し、ORAM や検索可能暗号では毎回のクエリごとにデータの更新と暗号文の再暗号化等が行われる。特に ORAM では格納位置のシャッフルも行われる。
- サーバへのクエリ内容：検索可能暗号では格納したデータがあるキーワードを含んでいるかどうかをクエリするのに対し、PIR と ORAM ではクライアントがアクセスしたいデータそのものについてクエリする。
- サーバに対して秘匿できる情報：いずれの技術もクエリ内容を秘匿する。特に、クエリに対してどのデータが返されたかに関する情報を秘匿する。しかし、検索可能暗号では検索に該当するデータの頻度などの一部の情報は秘匿しないのに対し、PIR と ORAM では頻度を含めた全ての情報を秘匿する。

ORAM は一般に検索可能暗号よりも強い安全性を保証できる代わりに効率性の理論限界が知られており、検索可能暗号ではサーバ自身もサーバに保存された内容を知り得ないという意味で PIR とは異なる。検索可能暗号では、サーバ上の暗号化された文書に対して、実用上問題ないと考えられる程度の多少の情報の漏洩を許しつつ、ORAM の理論限界を超えた効率的な検索機能を達成可能である。

■**検索可能暗号の分類と歴史** 検索可能暗号は、公開鍵暗号に基づく方式と共通鍵暗号に基づく方式に分類できる。公開鍵ベース方式 [75, 76] (Public-key Encryption with Keyword Search, PEKS) では公開鍵を用いて暗号化が可能であり、任意のクライアントがサーバに（暗号化）文書を格納可能となる。PEKS は Boneh ら [75] によって提案され、Abdalla ら [76] によって任意の匿名 ID ベース暗号方式を効率

的に PEKS に変換できることが示され、基本的な構成枠組みが完成した。以降、当初は考えられていなかった種々の攻撃に対する安全性 [77, 78] や複数キーワードを扱う方式 [79] 等の研究が進められている。一方で、共通鍵ベース方式 [80, 81] (Searchable Symmetric Encryption, SSE) では秘密鍵を知っている特定のクライアントしか (暗号化) 文書を格納できないものの、処理が高速であるという特長がある。特に、暗号文一つ一つと検索キーワードの一致を判定するアルゴリズムを考え、そして全ての暗号文にそのアルゴリズムを実行することを想定する PEKS とは異なり、SSE では暗号化データベース全体に対して検索を行うアルゴリズムを考えており、検索処理全体の効率性を意識して設計されている。そのような効率性の観点から、SSE に関する研究が現在の検索可能暗号研究の主流となっている。SSE はその概念自体が Song ら [82] によって提案された後、Curmola ら [80, 81] によってそのモデルと安全性が定式化され、具体的な方式も提案された。以降、SSE 方式の効率性改善だけでなく、理論限界の解析 [83, 84, 85] や、正しく結果を返さない不正なサーバを検知することで外部プロトコルと組み合わせた時の安全性を保証する方式 [86]、複数キーワードを扱う方式 [87]、一般的な情報検索を扱う方式 [88] 等の様々な研究が続けられているが、最も盛んに研究が進められているのは、暗号化データベースの動的な更新 (任意のタイミングでの文書ファイル・キーワードの追加/削除) を可能とする Dynamic SSE [89] である。実システム運用を考えると、暗号化データベースの更新機能は重要となる。

■(Dynamic) SSE 方式の特徴と比較 上で述べた通り、(Dynamic) SSE ではあらかじめ“漏洩を許す情報”を定め、検索処理や更新処理から“許した漏洩”以上の情報を漏らさないことを保証する。特に Dynamic SSE では、許す漏洩の程度によってフォワード安全性 [90] やバックワード安全性 [91] が明示的に定義される。前者は“文書がデータベースに追加された時、それ以前の検索処理からは追加された文書に関する情報がサーバに漏れない”ことを保証し、検索可能暗号本来の目的である“暗号化したまま検索を行う”ためには、フォワード安全性を満たすことが実用上重要である。後者のバックワード安全性は“文書がデータベースから削除された後、以降の検索処理から削除された文書の情報がサーバに漏れない”ことを保証し、近年最も盛んに研究されている Dynamic SSE の研究テーマである。しかし、その安全性の強さから効率的な実現が難しく構成が複雑になること、バックワード安全性が必要かどうかは応用先に強く依存すると考えられることから、具体的な構成については本ガイドラインでは割愛する。

表 3.1 にいくつかの代表的な Dynamic SSE 方式を示す。表中の記法は次の通り。 n はデータベース中の全文書ファイル数を表し、 N はデータベース中の (ファ

表 3.1 方式比較。方式名があるものは方式名を、方式名がない場合は著者名を記載している。計算量はいずれもクライアント側のものであり、◎は漸近的に最適である（原理上これ以上の効率化ができない）ことを表す。FP と BP はそれぞれフォワード安全性とバックワード安全性を表す。

	データベース サイズ	更新計算量	検索計算量	FP	BP	備考
[80, 81] ([92])	◎ : $\mathcal{O}(N)$	◎ : $\mathcal{O}(\mu)$	△ : $\mathcal{O}(n)$	✓	—	シンプルな構成
[90]	◎ : $\mathcal{O}(N)$	◎ : $\mathcal{O}(\mu)$	◎ : $\mathcal{O}(n_w)$	✓	—	公開鍵暗号技術が 構成に必要
[93]	◎ : $\mathcal{O}(N)$	◎ : $\mathcal{O}(\mu)$	◎ : $\mathcal{O}(n_w)$	✓	—	検索時に少し多く 情報漏洩を許す
[94]	○ : $\mathcal{O}(\tilde{N})$	○ : $\mathcal{O}(u_w)$	○ : $\mathcal{O}(u_w)$	✓	✓	比較的シンプルな 構成

イル識別子, キーワード) のペアの数である（データベースの表現については後述する）。通常、削除処理によってデータベースから削除ファイルに対応するペアが削除されるが、多くのバックワード安全な方式では当該ペアを削除せずに“当該ペアが削除された”という情報を新たにデータベースに追加することで削除したとみなす（検索時にクライアントがその情報を確認し、当該ペアを取り除くことで正しく検索結果を取り出すことができる）。そのため、データベースサイズは増大し、表中では $\tilde{N}$ ($\geq N$) で書く。 μ はその更新処理によってデータベースに追加/削除されるキーワードの数を表す。 n_w はデータベース中の検索キーワード w を含むファイル数を表し、 u_w はこれまでに w に対して行われてきた更新処理（追加及び削除）の総数を表す。すなわち、 $n_w \leq u_w$ が成り立つ。

なお、ほとんどの Dynamic SSE 方式においてクライアント側で秘密鍵以外の秘密情報を記憶しておく必要がある。そのような情報のサイズも比較対象ではあるものの、ここでは割愛する。

■代表的な (Dynamic) SSE 方式の紹介 具体的な検索可能暗号の実現方法として、代表的かつシンプルな方式を紹介する。なお、構成要素技術として、疑似ランダム関数 (Pseudo-Random Function, PRF) π を用いる。PRF π は、秘密鍵 k と任意の値 x を入力に取り、 y を出力する。この時、 y は同じ長さの乱数と見分けがつかず、疑似乱数として扱うことができる。また、データベースの記法についてもまとめる。識別子 id を持つ文書ファイル f_{id} に対し、形態素解析等を用いてキー

ワードを抽出した集合を $\mathcal{W}_{id}$ とする。ここで、識別子 id は文書ファイル f_{id} とは全く関係のない値（たとえば文書管理番号や文書ファイルのハッシュ値）であるとする。（Dynamic）SSE で考えるデータベースは、文書ファイル $(id, \mathcal{W}_{id})$ を基に作られた（ファイル識別子, キーワード）のペア (id, w) の集合とする。

以下では、Curtmola らによる SSE 方式である SSE-2 [80, 81] を紹介する。より正確には、オリジナルの方式 [80, 81] には誤り及び冗長な点が含まれるため、その点を修正し、かつ Dynamic SSE に拡張した方式 [92]（表 3.1 の一つ目の方式）を示す。なお、本方式はフォワード安全性を満たす。

- セットアップ及びデータベース生成：クライアントは、PRF 鍵 k をランダムに選び、空の配列 Index を暗号化データベースとして用意しておく。
- 文書ファイルの追加：追加したい文書ファイルを $f_{id} = (id, \mathcal{W}_{id})$ とする。クライアントは、全ての $w \in \mathcal{W}_{id}$ に対して、 $\text{addr} := \pi(k, w \| id)$ を計算する。クライアントは計算した全てのアドレスと識別子 id をサーバに送り、サーバは全てのアドレス addr に対し、 $\text{Index}[\text{addr}]$ に id を格納し、暗号化データベースを更新する。クライアントは別途追加した id を記憶しておく。
- 文書ファイルの削除：削除したい文書ファイルの識別子 id に対し、クライアントは id をサーバに送る。サーバは Index の格納値を調べ、 id が格納されているアドレスを全て空にし、暗号化データベースを更新する。クライアントは記憶していた id を削除する。
- 検索：検索したいキーワード q に対し、クライアントは記憶している全ての識別子 id に対して、トラップドア $\tau_{id} := \pi(k, q \| id)$ を計算し、計算したトラップドアを全てサーバに送る。サーバは、受け取った各 τ_{id} に対して、 $\text{Index}[\tau_{id}]$ の格納値を確認する。もし何かしらの値（すなわち id ）が格納されていれば、それを集合 $\mathcal{X}_q$ に含める（ $\mathcal{X}_q$ は最初に空集合として初期化しておく）。最終的にサーバはクライアントに $\mathcal{X}_q$ を渡し、クライアントは $\mathcal{X}_q$ を検索結果として受理する。

上記の構成において、文書ファイルの追加をセットアップ時のみに行い、それ以降追加も削除も行わなければ、Dynamic ではない通常の SSE 方式となる。

参考文献

- [1] A. Shamir: “How to Share a Secret”, Commun. ACM, **22**, 11, pp. 612–613 (1979).
- [2] G. R. Blakley: “Safeguarding cryptographic keys”, MARK 1979, pp. 313–

317 (1979).

- [3] M. Itoh, A. Saito and T. Nishizeki: “Secret Sharing Scheme Realizing General Access Structure”, IEEE Globecom 1987, pp. 99–102 (1987).
- [4] B. Chor, G. Goldwasser, S. Micali and B. Awerbuch: “Verifiable Secret Sharing Simultaneity in the Presence of Faults”, FOCS 1985, ACM, pp. 383–395 (1985).
- [5] M. Ben-Or, S. Goldwasser and a. Wigderson: “Completeness Theorems for Non-Cryptographic Fault Tolerant Distributed Computation”, STOC 1988, ACM, pp. 1–10 (1988).
- [6] M. Naor and A. Shamir: “Visual Cryptography”, EUROCRYPT 1994, Springer-Verlag, pp. 1–12 (1994).
- [7] M. Hillery, V. Buzěk and A. Berthiaume: “Quantum Secret Sharing”, Los Alamos e-print archive, **quant-ph/9806063**, (1998).
- [8] R. Cleve, D. Gottesman and H.-K. Lo: “How to Share a Quantum Secret”, Physical Review Letters, **83**, 3, pp. 648–651 (1999).
- [9] A. Shamir: “How to share a secret”, Commun. ACM, **22**, 11, pp. 612–613 (1979).
- [10] E. Karnin, J. Greene and M. E. Hellman: “On Secret Sharing Systems”, IEEE Trans. Information Theory, **29**, 1, pp. 35–41 (1983).
- [11] S. Martín, C. Padró and A. Yang: “Secret Sharing, Rank Inequalities and Information Inequalities”, CRYPTO 2013, Springer-Verlag, pp. 277–288 (2013).
- [12] M. Karchmer and A. Wigderson: “On Span Programs”, Annual Structure in Complexity Theory Conference, pp. 102–111 (1993).
- [13] G. R. Blakley and C. Meadows: “Security of ramp schemes”, CRYPTO 1984, Springer-Verlag, pp. 242–268 (1984).
- [14] 山本: “ (k, L, n) しきい値秘密分散システム”, 電子通信学会論文誌, **J68-A**, 9, pp. 945–952 (1985).
- [15] H. Krawczyk: “Secret sharing made short”, CRYPTO 1993, Springer-Verlag, pp. 136–146 (1993).
- [16] S. Shin, S. Yamada, G. Hanaoka, Y. Ishida, A. Kunii, J. Oketani, S. Kunii and K. Tomomura: “How to Extend CTRT for AES-256 and AES-192”, IEICE Trans., **E105-A(8)**, pp. 1121–1133 (2022).
- [17] “ZENMU-AONT に関する論文が 国際会議で best paper premium award を

- 受賞”, <https://www.atpress.ne.jp/news/191661>.
- [18] “ISO/IEC 19592-2:2017 Information technology - Security techniques - Secret sharing - Part 2: Fundamental mechanisms”.
 - [19] M. Iwamoto and H. Yamamoto: “Strongly Secure Ramp Secret Sharing Schemes for General Access Structures”, *Information Processing Letters*, **97**, 2, pp. 52–57 (2006).
 - [20] J. Benaloh and J. Leichter: “Generalized Secret Sharing and Monotone Functions”, *CRYPTO 1988*, Springer New York, pp. 27–35 (1988).
 - [21] G. S. Vernam: “Cipher Printing Telegraph Systems for Secret Wire and Radio Telegraphic Communications”, *J. of American Institute for Electrical Engineering*, **45**, pp. 109–115 (1926).
 - [22] D. Beaver, S. Micali and P. Rogaway: “The Round Complexity of Secure Protocols (Extended Abstract)”, *STOC 1990*, ACM, pp. 503–513 (1990).
 - [23] Y. Lindell, B. Pinkas, N. P. Smart and A. Yanai: “Efficient Constant-Round Multi-party Computation Combining BMR and SPDZ”, *J. of Cryptology*, **32**, 3, pp. 1026–1069 (2019).
 - [24] O. Goldreich, S. Micali and A. Wigderson: “How to Play any Mental Game or A Completeness Theorem for Protocols with Honest Majority”, *STOC 1987*, ACM, pp. 218–229 (1987).
 - [25] M. Ben-Or, O. Goldreich, S. Goldwasser, J. Håstad, J. Kilian, S. Micali and P. Rogaway: “Everything Provable is Provable in Zero-Knowledge”, *CRYPTO 1988*, Springer-Verlag, pp. 37–56 (1988).
 - [26] T. Araki, J. Furukawa, Y. Lindell, A. Nof and K. Ohara: “High-Throughput Semi-Honest Secure Three-Party Computation with an Honest Majority”, *ACM CCS 2016*, pp. 805–817 (2016).
 - [27] N. Attrapadung, G. Hanaoka, T. Matsuda, H. Morita, K. Ohara, J. C. N. Schuldt, T. Teruya and K. Tozawa: “Oblivious Linear Group Actions and Applications”, *ACM CCS 2021*, pp. 630–650 (2021).
 - [28] Y. Lindell and A. Nof: “A Framework for Constructing Fast MPC over Arithmetic Circuits with Malicious Adversaries and an Honest-Majority”, *ACM CCS 2017*, pp. 259–276 (2017).
 - [29] R. Cramer, I. Damgård and Y. Ishai: “Share Conversion, Pseudorandom Secret-Sharing and Applications to Secure Computation”, *TCC 2005*, Springer-Verlag, pp. 342–362 (2005).

- [30] NTT: “秘密計算基盤 (Trust-SC) V3.0 「算師 (R)」 | NTT R&D Website”. <https://www.rd.ntt/research/SI0005.html>.
- [31] R. Kikuchi, D. Ikarashi, T. Matsuda, K. Hamada and K. Chida: “Efficient Bit-Decomposition and Modulus-Conversion Protocols with an Honest Majority”, ACISP 2018, Springer-Verlag, pp. 64–82 (2018).
- [32] D. Beaver: “Efficient Multiparty Protocols Using Circuit Randomization”, CRYPTO 1991, Springer-Verlag, pp. 420–432 (1991).
- [33] I. Damgård, V. Pastro, N. P. Smart and S. Zakarias: “Multiparty Computation from Somewhat Homomorphic Encryption”, CRYPTO 2012, Springer-Verlag, pp. 643–662 (2012).
- [34] N. P. Smart and T. Tanguy: “TaaS: Commodity MPC via Triples-as-a-Service”, CCSW@CCS 2019, ACM, pp. 105–116 (2019).
- [35] ZenmuTech: “データを秘匿化したま計算 | ZenmuTech”. <https://zenmotech.com/products/secure-computation/>.
- [36] 石田, 國井, 桶谷, 大畑, 松田, アッタラパドゥン, 花岡: “Query Ahead: 平易な記述が可能な秘匿 DB クエリーシステムの設計と実装”, SCIS 2020 (2020).
- [37] A. C. Yao: “How to Generate and Exchange Secrets (Extended Abstract)”, FOCS 1986, IEEE Computer Society, pp. 162–167 (1986).
- [38] M. Bellare, V. T. Hoang and P. Rogaway: “Foundations of Garbled Circuits”, ACM CCS 2012, pp. 784–796 (2012).
- [39] M. O. Rabin: “How To Exchange Secrets with Oblivious Transfer”. ePrint 187/2005.
- [40] S. Zahur, M. Rosulek and D. Evans: “Two Halves Make a Whole - Reducing Data Transfer in Garbled Circuits Using Half Gates”, EUROCRYPT 2015, Springer-Verlag, pp. 220–250 (2015).
- [41] V. Kolesnikov and T. Schneider: “Improved Garbled Circuit: Free XOR Gates and Applications”, ICALP 2008, Springer-Verlag, pp. 486–498 (2008).
- [42] M. Naor, B. Pinkas and R. Sumner: “Privacy preserving auctions and mechanism design”, ACM EC 1999, pp. 129–139 (1999).
- [43] B. Pinkas, T. Schneider, N. P. Smart and S. C. Williams: “Secure Two-Party Computation Is Practical”, ASIACRYPT 2009, Springer-Verlag, pp. 250–267 (2009).
- [44] S. Goldwasser, S. Micali and C. Rackoff: “The Knowledge Complexity of Interactive Proof Systems”, SIAM J. Computing, **18**, 1, pp. 186–208 (1989).

- [45] B. Manuel, Feldman and P. M. Silvio: “Non-Interactive Zero-Knowledge and Its Applications”, STOC 1988, ACM, pp. 103–112 (1988).
- [46] C.-P. Schnorr: “Efficient Identification and Signatures for Smart Cards”, CRYPTO 1989, Springer-Verlag, pp. 239–252 (1989).
- [47] R. Gennaro, C. Gentry, B. Parno and M. Raykova: “Quadratic Span Programs and Succinct NIZKs without PCPs”, EUROCRYPT 2013, Springer-Verlag, pp. 626–645 (2013).
- [48] B. Parno, J. Howell, C. Gentry and M. Raykova: “Pinocchio: Nearly Practical Verifiable Computation”, IEEE S&P 2013 (2013).
- [49] J. Groth: “On the Size of Pairing-Based Non-interactive Arguments”, EUROCRYPT 2016, Springer-Verlag, pp. 305–326 (2016).
- [50] E. Ben-Sasson, I. Bentov, Y. Horesh and M. Riabzev: “Scalable Zero Knowledge with No Trusted Setup”, CRYPTO 2019, Springer-Verlag, p. 701–732 (2019).
- [51] B. Bünz, J. Bootle, D. Boneh, A. Poelstra, P. Wuille and G. Maxwell: “Bulletproofs: Short Proofs for Confidential Transactions and More”, IEEE S&P 2018, pp. 315–334 (2018).
- [52] O. Goldreich: “Towards a Theory of Software Protection and Simulation by Oblivious RAMs”, STOC 1987, ACM, pp. 182–194 (1987).
- [53] R. Ostrovsky: “Efficient computation on oblivious RAMs”, STOC 1990, ACM, pp. 514–523 (1990).
- [54] O. Goldreich and R. Ostrovsky: “Software protection and simulation on oblivious RAMs”, **43 (3)**, pp. 431–473 (1996).
- [55] E. Shi, T.-H. H. Chan, E. Stefanov and M. Li: “Oblivious RAM with $O((\log N)^3)$ worst-case cost”, ASIACRYPT 2011, Springer-Verlag, pp. 197–214 (2011).
- [56] E. Kushilevitz, S. Lu and R. Ostrovsky: “On the (In)Security of Hash-Based Oblivious RAM and A New Balancing Scheme”, SODA 2012, pp. 143–156 (2012).
- [57] E. Stefanov, M. van Dijk, E. Shi, C. W. Fletcher, L. Ren, X. Yu and S. Devadas: “Path ORAM: an extremely simple oblivious RAM protocol”, ACM CCS 2013, pp. 299–310 (2013).
- [58] S. Patel, G. Persiano, M. Raykova and K. Yeo: “PanORAMa: Oblivious RAM with Logarithmic Overhead”, FOCS 2018, ACM, pp. 871–882 (2018).

- [59] G. Asharov, I. Komargodski, W.-K. Lin, K. Nayak, E. Peserico and E. Shi: “OptORAMa: Optimal Oblivious RAM”, EUROCRYPT (2) 2020, Springer-Verlag, pp. 403–432 (2020).
- [60] G. Asharov, I. Komargodski, W.-K. Lin and E. Shi: “Oblivious RAM with Worst-Case Logarithmic Overhead”, CRYPTO (4) 2021, Springer-Verlag, pp. 610–640 (2021).
- [61] S. Lu and R. Ostrovsky: “Distributed Oblivious RAM for Secure Two-Party Computation”, TCC 2013, Springer-Verlag, pp. 377–396 (2013).
- [62] E. Boyle, K.-M. Chung and R. Pass: “Oblivious parallel ram and applications”, TCC 2016, Springer-Verlag, pp. 175–204 (2016).
- [63] B. Chor, E. Kushilevitz, O. Goldreich and M. Sudan: “Private Information Retrieval”, FOCS, IEEE, pp. 41–50 (1995).
- [64] B. Chor, E. Kushilevitz, O. Goldreich and M. Sudan: “Private Information Retrieval”, J. ACM, **45**, 6, pp. 965–981 (1998).
- [65] E. Kushilevitz and R. Ostrovsky: “Replication Is Not Needed: Single Database, Computationally-Private Information Retrieval”, FOCS 1997, IEEE, pp. 364–373 (1997).
- [66] H. Corrigan-Gibbs and D. Kogan: “Private Information Retrieval with Sublinear Online Time”, EUROCRYPT 2020, Springer-Verlag, pp. 44–75 (2020).
- [67] H. Corrigan-Gibbs, A. Henzinger and D. Kogan: “Single-Server Private Information Retrieval with Sublinear Amortized Time”, EUROCRYPT 2022, Springer International Publishing, pp. 3–33 (2022).
- [68] L. Holzbaur, C. Hollanti and A. Wachter-Zeh: “Computational Code-Based Single-Server Private Information Retrieval”, IEEE ISIT 2020, pp. 1065–1070 (2020).
- [69] G. N. Alfarano, K. Khathuria and V. Weger: “A Survey on Single Server Private Information Retrieval in A Coding Theory Perspective”, Applicable Algebra in Engineering, Communication and Computing (2021).
- [70] H. Sun and S. A. Jafar: “The Capacity of Private Information Retrieval”, IEEE Trans. Information Theory, **63**, 7, pp. 4075–4088 (2017).
- [71] D. Augot, F. Levy-dit Vehel and A. Shikfa: “A Storage-Efficient and Robust Private Information Retrieval Scheme Allowing Few Servers”, CANS 2014, Springer International Publishing, pp. 222–239 (2014).

- [72] J. Katz and L. Trevisan: “On The Efficiency of +Local Decoding Procedures for Error-Correcting Codes”, STOC 2000, ACM, pp. 80–86 (2000).
- [73] A. Beimel and Y. Stahl: “Robust Information-Theoretic Private Information Retrieval”, J. of Cryptology, **20**, 3, pp. 295–321 (2007).
- [74] K. Kurosawa: “How to Correct Errors in Multi-server PIR”, ASIACRYPT 2019, Springer-Verlag, pp. 564–574.
- [75] D. Boneh, G. Di Crescenzo, R. Ostrovsky and G. Persiano: “Public Key Encryption with Keyword Search”, EUROCRYPT 2004, Springer-Verlag, pp. 506–522 (2004).
- [76] M. Abdalla, M. Bellare, D. Catalano, E. Kiltz, T. Kohno, T. Lange, J. Malone-Lee, G. Neven, P. Paillier and H. Shi: “Searchable Encryption Revisited: Consistency Properties, Relation to Anonymous IBE, and Extensions”, J. of Cryptology, **21**, 3, pp. 350–391 (2007).
- [77] Q. Tang: “Towards Forward Security Properties for PEKS and IBE”, Information Security and Privacy, ACISP 2015, Springer-Verlag, pp. 127–144 (2015).
- [78] R. Chen, Y. Mu, G. Yang, F. Guo and X. Wang: “A New General Framework for Secure Public Key Encryption with Keyword Search”, Information Security and Privacy, ACISP 2015, Springer-Verlag, pp. 59–76 (2015).
- [79] P. Golle, J. Staddon and B. Waters: “Secure Conjunctive Keyword Search over Encrypted Data”, ACNS 2004, Springer-Verlag, pp. 31–45 (2004).
- [80] R. Curtmola, J. Garay, S. Kamara and R. Ostrovsky: “Searchable Symmetric Encryption: Improved Definitions and Efficient Constructions”, ACM CCS 2006, pp. 79–88 (2006).
- [81] R. Curtmola, J. A. Garay, S. Kamara and R. Ostrovsky: “Searchable Symmetric Encryption: Improved Definitions and Efficient Constructions”, J. of Computer Security, **19**, 5, pp. 895–934 (2011).
- [82] D. X. Song, D. Wagner and A. Perrig: “Practical Techniques for Searches on Encrypted Data”, IEEE S&P 2000, pp. 44–55 (2000).
- [83] G. Asharov, G. Segev and I. Shahaf: “Tight Tradeoffs in Searchable Symmetric Encryption”, J. of Cryptology, **34**, 9, pp. 1–37 (2021).
- [84] G. Asharov, M. Naor, G. Segev and I. Shahaf: “Searchable Symmetric Encryption: Optimal Locality in Linear Space via Two-Dimensional Balanced Allocations”, SIAM J. Computing, **50**, 5, pp. 1501–1536 (2021).

- [85] D. Cash and S. Tessaro: “The Locality of Searchable Symmetric Encryption”, EUROCRYPT 2014, Springer-Verlag, pp. 351–368 (2014).
- [86] K. Kurosawa and Y. Ohtaki: “UC-Secure Searchable Symmetric Encryption”, Financial Cryptography and Data Security, FC 2012, Springer-Verlag, pp. 285–298 (2012).
- [87] D. Cash, S. Jarecki, C. Jutla, H. Krawczyk, M.-C. Roşu and M. Steiner: “Highly-Scalable Searchable Symmetric Encryption with Support for Boolean Queries”, CRYPTO 2013, Springer-Verlag, pp. 353–373 (2013).
- [88] M. Chase and S. Kamara: “Structured Encryption and Controlled Disclosure”, ASIACRYPT 2010, Springer-Verlag, pp. 577–594 (2010).
- [89] S. Kamara, C. Papamanthou and T. Roeder: “Dynamic Searchable Symmetric Encryption”, ACM CCS 2012, pp. 965–976 (2012).
- [90] R. Bost: “ $\Sigma\phi\phi\phi$: Forward Secure Searchable Encryption”, ACM CCS 2016, pp. 1143–1154 (2016).
- [91] R. Bost, B. Minaud and O. Ohrimenko: “Forward and Backward Private Searchable Encryption from Constrained Cryptographic Primitives”, ACM CCS 2017, pp. 1465–1482 (2017).
- [92] Y. Watanabe, T. Nakai, K. Ohara, T. Nojima, Y. Liu, M. Iwamoto and K. Ohta: “How to Make a Secure Index for Searchable Symmetric Encryption, Revisited”, IEICE Trans., **E105-A(12)**, pp. 1559–1577 (2022).
- [93] M. Etemad, A. Küpçü, C. Papamanthou and D. Evans: “Efficient Dynamic Searchable Encryption with Forward Privacy”, Privacy Enhancing Technologies (PoPETs), Vol. 2018(1), pp. 5–20 (2018).
- [94] J. G. Chamani, D. Papadopoulos, C. Papamanthou and R. Jalili: “New Constructions for Forward and Backward Private Symmetric Searchable Encryption”, ACM CCS 2018, pp. 1038–1055 (2018).

第 4 章

おわりに

本ガイドラインに記載した内容は、2022 年 9 月以前に公開された情報に基づいている。2022 年 9 月以降も多くの高機能暗号に関連する情報が、各社の広報、学会、標準化団体において公開されてきているが、発表されたばかりであり、安全性の評価が完全には行われていない方式もある。本ガイドラインに記載した高機能暗号は、2022 年 9 月時点で、主要国際学会で発表されており、有力な攻撃法が発見されておらず、かつ、十分な性能を持つと考えられる方式を選んでいる。

高機能暗号の特徴であるが、機能が異なれば、異なる暗号となる。このため、多くの種類の高機能暗号があり、それらすべて本ガイドラインに掲載することはできない。主要と考えられる高機能暗号に絞って掲載することになった。

また、高機能暗号は多種多様であるが故に、導入するためには専門的な知識が要求される。この知識を有し、かみ砕いて説明でき、ケースに応じて応用するスキルを有する人材の確保は課題となっている。さらに、既存システムとの整合性を考慮し、例えば、組織内の端末にソフトウェアとして導入できるか、マネジメントシステムと連携できるか、Web ブラウザ上で動作ができるか、インターフェースはどのような形態がよいか等、これらは実際のシステム開発者や利用者と密に連携をとりつつ開発することが必要となる。

本ガイドラインが、用途に適した高機能暗号を選択するため、および、研究者、システム開発者、利用者にとっての指針となれば幸いである。

CRYPTREC 暗号技術ガイドライン（高機能暗号）

[CRYPTREC GL-2005-2022]

不許複製 禁無断転載

発行日：2023 年 3 月 31 日（第 1 版）

発行者

- 〒184-8795

東京都小金井市貫井北町四丁目 2 番 1 号

国立研究開発法人 情報通信研究機構

（サイバーセキュリティ研究所 セキュリティ基盤研究室）

NATIONAL INSTITUTE OF INFORMATION AND COMMUNICATIONS TECHNOLOGY

4-2-1 NUKUI-KITAMACHI, KOGANEI

TOKYO, 184-8795 JAPAN

- 〒113-6591

東京都文京区本駒込二丁目 2 8 番 8 号

独立行政法人情報処理推進機構

（セキュリティセンター セキュリティ技術評価部 暗号グループ）

INFORMATION-TECHNOLOGY PROMOTION AGENCY, JAPAN

2-28-8 HONKOMAGOME, BUNKYO-KU

TOKYO, 113-6591 JAPAN