東海大学大学院　令和５年度博士論文

複数組織対応属性ベース暗号を用いた研究データ向けファイル共有システムに関する研究

指導　　大東　俊博　教授

東海大学大学院　総合理工学研究科
総合理工学専攻
石橋　拓哉

内容梗概

近年，様々なデータの管理にオンラインストレージが用いられることが増えてきており，個人でのデータのバックアップのみでなく，企業など組織での会議資料の保存・共有にも用いられている．オンラインストレージ単体でデータの管理を行う場合，アクセス制御やサービスの提供者によるデータの暗号化は行われているが，あくまでサービス提供者による処理となっている．そのため，サービス提供者側に悪意のある管理者がいた場合には，保存しているデータの流出等の危険性が考えられる．したがって，これらのオンラインストレージなどを利用する場合には，ユーザ側で暗号化を行い，データを保護するシステムが必要となる．この問題に対応する既存研究として，属性ベース暗号を用いたファイル共有システムの提案が行われている．属性ベース暗号は公開鍵暗号の一種であり，暗号方式単体でデータの暗号化および暗号化したデータへのアクセス制御を実現している．しかしながら，このファイル共有システムに用いられている属性ベース暗号は単一組織での使用が想定されている方式となっているため，複数組織間などでデータの共有を行う場合などには使用することができない．そこで本研究では，ファイルの共有・管理を複数組織で行うシステムを設計することを目的とする．また，既存研究の属性ベース暗号を用いたファイル共有システムを参考に，複数組織間で利用可能な属性ベース暗号 (MA-ABE) を用いたシステム設計を提案する．

最初に，本研究で提案するファイル共有システムについて説明する．提案システムに必要な要件を確認・整理した後，提案システムを実現するために最適な MA-ABE の方式の選定を行う．現在まで，MA-ABE の方式は複数提案されているが，MA-ABE を実際のシステムとして利用することを想定し，実装を行い，処理時間などを含めた評価を行っている研究はされていない．そこで様々な MA-ABE の方式の特徴を整理・比較を示し，処理時間の計測・評価を行った上で選定する．その後，選定した MA-ABE の方式を用いた提案システムを実際に利用する際に生じる，ユーザの鍵発行や，データのアップロード・ダ

ウンロード時の処理時間の計測・評価を行う．

次に，提案システムを実際に運用する際に詳細な設計が必要となる，各組織における属性管理の方法，保存データに関するファイル名・ディレクトリ名の管理方法，アップロード時などの処理を制御するアップロードマネージャの運用・設置方法，鍵発行機関である KGC の運用方法について検討し，運用方法として最適な方法の考察・設計を行う．その後，提案システムを運用する際に発生するシステムの処理部分である，ユーザの鍵発行，データのダウンロード・アップロード時の処理の手順に関して，実環境上で構築するための詳細な設計を示す．また，実際のユースケースに基づいて，設計した処理の検証を行い，システムの要件を満たす設計であることを確認する．さらに，今回行った詳細な設計を用いて，提案システムを実現できる MA-ABE の方式が満たすべき条件に関して考察する．

以上のように本研究では，複数組織間で利用可能な属性ベース暗号を用いた研究データ向けのファイル共有システムの提案・設計を行う．本研究によって，今まで議論がされていない MA-ABE を用いるファイル共有システムに関して，実際に運用するために必要となる要件およびそれを満たす方法，システムを実装する際に必要となる設計を示し，提案システムの開発およびその実現可能性を明確化する．

目次

図目次

表目次

第 1 章

序論

1.1 研究背景

近年，Dropbox[*1] に代表されるオンラインストレージサービスの普及により，個人でのデータのバックアップや，企業などのデータや会議資料等の管理にオンラインストレージが用いられることが増えてきており，オンラインでのファイル共有は DX には欠かせないものとなっている．オンラインストレージサービスではストレージの管理者により，アクセス制御やディスクごとに暗号化がされていたりするが，これらは全てサービスの提供者側によって実施されているものであり，悪意のある管理者がいた場合データを覗き見られる危険性がある．そこでこれらのオンラインストレージを利用する場合には，ユーザ側で暗号化を行い，データを保護するシステムが必要であると考えられる．

利用者が自らデータを保護する方法として，ユーザ側でデータを暗号化する VeraCrypt[*2] などのシステムが既に存在する．しかし，これらは共通鍵暗号などを利用してコンテンツの暗号化を行うものが多く，データを共有したいグループごとに使用する鍵を変える必要がある．その結果，ユーザはデータを共有したいグループの数と同数の鍵の管理を行う必要があり，鍵の配布・管理のコストが大きくなってしまう．これらのデメリットを補う，公開鍵暗号の一種である暗号文ポリシー属性ベース暗号（Ciphertext-Policy Attribute-Based Encryption: CP-ABE）[4] が提案されている．CP-ABE は属性値 (ID・所属・役職など) の論理式で表現されたアクセスポリシー（以下，アクセス権）を暗号文に埋め込み，その暗号文をアクセス権を満たす属性を有したユーザの秘密鍵でしか，その暗号文を復号できな

[*1] https://www.dropbox.com

[*2] https://www.veracrypt.fr/en/Home.html

くすることで，きめ細やかなアクセス制御機能を暗号化処理に付加できる．CP-ABE ではユーザは鍵発行センター (Key Generation Center: KGC) にて，自身の属性が含まれた秘密鍵を生成し，適切な認証を経て取得することで閲覧権限があるデータを復号できるようになる．

ファイル共有システムは属性ベース暗号の最も期待される応用先と言える．そのため，CP-ABE を使った様々なファイル共有システムに関して先行研究が行われている．例として，CP-ABE を用いることでオンラインストレージ上のファイルの閲覧権限を柔軟に制御するシステム [5][6] や，医療機関での電子カルテを担当医のみが閲覧可能なものから，緊急時に全スタッフが閲覧可能となるような変更を行える共有システム [7]，従来のファイル共有サービスと異なり，コンテンツだけでなくファイル名・ディレクトリ名を含むディレクトリ構造全体を暗号化し，ファイル名・ディレクトリ名の秘匿および編集権限の制御を行うシステム [2] などが提案されている．これらのシステムで使用している CP-ABE は単一の組織での使用を前提としている．そのため，ファイル共有システム自体が単一組織での利用に限られる．ファイル共有システムを実際に利用する場面を考えた場合，共同研究などを行っている際に研究データや論文の共有など，複数組織間でシステムを利用したい場面が出てくることが想定される．しかし，既存研究の属性ベース暗号を用いたファイル共有システムでは，複数組織間でデータを共有するなどの用途には使用できず，これらのニーズに対応することができない．

複数組織間での研究データの共有が行えるサービスとして，Gakunin RDM[*3] といったサービスが存在する．このサービスは国立情報学研究所[*4] によって提供されており，外部のオンラインストレージやツールなどと連携を行い，研究データの管理・共有を柔軟に行えるようになっている．しかしながら，このサービスでは既存のオンラインストレージなどを用いてファイルの管理を行うため，アクセス制御を行うことは可能であるものの，ファイルの暗号化などはされていない．そのため，安全に研究データの管理を行うためには，複数組織間でのファイル共有が可能であり，更にユーザ側でファイルを暗号化して管理を行うシステムの実現が必要となる．

以上のような背景から本論文では，複数組織間で利用可能なファイル共有システムを実現させるための研究を行う．

[*3] https://rcos.nii.ac.jp/service/rdm/
[*4] https://www.nii.ac.jp/

1.2 研究目的

本研究では，複数組織間で利用可能なファイル共有システムを実現することを目的とする．まず，複数組織間で利用可能なファイル共有システムを実現するために必要となる，システムの要件の定義をする．次に，定義した要件を満たす複数組織間で利用可能なファイル共有システムの具体的な構成方法を提案し，それに伴って必要となる検討・考察を行う．その後，提案した構成方法を元に実際のユースケースを想定したシステムの処理手順の詳細設計を示し，実際のユースケースに沿って設計した処理手順を検証する．さらに，提案システムが複数組織間で利用可能なファイル共有システムの要件を満たすことを確認し，本研究の提案方法にて，複数組織間で利用可能なファイル共有システムが実現可能であることを明確化する．

複数組織間で利用可能なファイル共有システムを実現する方法として，本研究では複数組織で利用可能な属性ベース暗号 (Multi-Authority Attribute-Based Encryption: MA-ABE) を用いる方法を提案する．MA-ABE は複数の提案がされている [8][9][10][11] [12][13] が，実用的なファイル共有のユースケースを十分に考慮した，システムの提案・評価および考察などは行われていない．そのため，暗号方式を実際に使用する際の鍵や属性などの具体的な管理手法の検討，およびそれらの情報の量のユースケースに基づいた具体的な検討やその量の許容範囲の確認，KGC などの信頼が必要となる機関の具体的な構築方法の確立，暗号方式自体ではサポートされていないが，実システムとして運用する際に使用している情報の更新や失効などが発生する可能性，および具体的かつ技術的な対処方法の検討，暗号方式の中で用いられているパラメータに具体的にどのような値を用いるのか，複数組織間でやり取りを行う情報の具体的な受け渡し方法，またそのデータのフォーマットなど，実システムとして運用する場合には方式をそのまま用いるだけでは足りず，検討・考察を行うべきことが多数存在する．そこで，実システムとして運用可能とための，MA-ABE を用いたファイル共有システムの設計・提案を行い，その評価と考察を行う．

本研究の目的を達成するため，まず MA-ABE を用いたファイル共有システムの提案を行い，その評価・考察をする．最初に，実際のユースケースを考慮したファイル共有システムを提案する．そして，先行研究にて提案されている MA-ABE の方式を分類・比較し，提案システムに最適な MA-ABE の選定を行う．次に，選定を行った MA-ABE の各アルゴリズムの通信時間などを含めた，実際の利用シーンを想定した処理時間を計測する．こ

れにより提案方式の定量的評価を行い，提案するファイル共有システムが実用可能であることを示す．さらに，提案するシステムを実際に運用する際に，ユーザの人事異動，卒業，退学などによる属性情報の変更が発生することが想定される．そのため，これらの場合に想定される鍵失効に関する問題，およびその解決方法についての考察も行う．

その次に，具体的なファイル共有システムとして実運用する際に必要となる，システムの処理手順や，様々なフォーマットなどに関して設計する．その上で実運用を考慮した際に生じると考えられる，提案システムにおける組織間での属性の取り扱いに関する問題点や，ファイルの閲覧を制御するリストファイルの運用方法，ファイルの編集権限を制御するために必要となるアップロードマネージャの設置方法や，KGC の運用方法に関する考察・設計なども行う．

本論文により，実運用を十分に考慮した複数組織間で利用可能なファイル共有システムの具体的な設計・提案を行い，提案システムの実運用時の視点における，運用課題の抽出やシステム構築方法を考察する．以上の提案・考察により，安全かつ実用的なファイル共有システムの実運用に関する方法を示し，提案システムの実現性を明確化する．

1.3　論文構成

本論文は 5 章から構成されている．第 2 章では，本研究に関連する先行研究と関連技術について述べる．その後，本研究で提案を行う複数組織で利用可能なファイル共有システムに関する要件を定め概説する．第 3 章では，第 2 章で定義した要件を満たす，複数組織間で利用可能なファイル共有システムの提案・考察をする．まず初めに，提案システムに必要な要件を満たす方法として，MA-ABE を用いた複数組織間で利用可能なファイル共有システムの提案を行い，提案システムに用いる MA-ABE の方式を選定するため，先行研究にて提案されている MA-ABE の比較・評価を実施する．次に，提案システムに用いる MA-ABE の方式の詳細な計測・評価をする．最後に，提案システムを運用する際に生じると考えられる問題点に関して考察を行う．第 4 章では，第 3 章にて提案したファイル共有システムに関して，実運用するための詳細な設計を行う．提案システムにおける組織間での属性の取り扱いに関する問題点，ファイルの閲覧を制御するリストファイルの運用方法，ファイルの編集権限を制御するために必要となるアップロードマネージャの設置方法，KGC の運用方法に関する考察・設計など，第 2 章にて定めた要件を満たすより詳細な方法の考察・設計をする．その後，実際の処理に関する設計を示し，それらの設計通り

に処理を行うようにすることで，提案システムが満たすべき要件を全て満たしていることを確認する．また，第 3 章にて選定を行った MA-ABE の方式以外の方式を用いて，提案システムを構築する場合の考察を行う．最後に，第 5 章では結論として本論文の研究成果をまとめる．また，本論文執筆における引用文献および参考資料などを参考文献の項目に掲載する．

本論文の第 3 章の一部は文献 [1] を基に構成されており，第 4 章の一部は文献 [3] を基に構成されている．

第 2 章

先行研究と関連技術

2.1　緒言

本章では，本研究に関連する先行研究と関連技術について述べる．まず，第 2.2 節では公開鍵暗号の概説を行い, 第 2.3 節では公開鍵暗号の一種である ID ベース暗号について説明する．次に，第 2.4 節では先行研究で使用されている単一組織での使用が想定されている属性ベース暗号の説明を述べる．さらに，第 2.5 節では既存のファイル共有システムとして，属性ベース暗号の一種である CP-ABE を用いたファイル共有システムなどについて紹介する．その後，既存のファイル共有システムでは対応していない，複数組織で利用可能なファイル共有システムが必要であることを説明し，複数組織で利用可能なファイル共有システムを運用する際に満たすべき要件について整理し概説する．

2.2　公開鍵暗号

暗号とは，元となるデータなどの平文に対して特別な処理を施し，第三者から平文の内容を保護するための技術である．この平文に処理を施したデータを暗号文と呼び，平文から暗号文に変換する処理を暗号化，暗号文を平文に変換する処理を復号と呼ぶ．またその時に使用する鍵をそれぞれ，暗号化鍵，復号鍵と呼ぶ．暗号方式は大きく共通鍵暗号と，公開鍵暗号の 2 つの方式に分類される．

共通鍵暗号とは，暗号化鍵と復号鍵に共通の鍵を用いる方式である．そのため事前にデータの送信者と受信者で鍵の共有を行って置く必要がある．共通鍵暗号は公開鍵暗号に比べて高速に処理が行えるといった特徴がある．代表的な共通鍵暗号方式として AES[14]

などがある．公開鍵暗号とは，暗号化鍵と復号鍵がそれぞれ別の鍵が用いられる方式である．暗号化鍵と復号鍵のペアを作成し，暗号化鍵を公開し（公開鍵），復号鍵は秘密情報（秘密鍵）とし自身で管理を行う．このように公開鍵暗号では，暗号化するための鍵を公開することができるため，事前に鍵を共有する必要がない．代表的な公開鍵暗号方式として RSA 暗号 [15] や楕円曲線暗号 [16] などがある．

公開鍵暗号は共通鍵暗号と比べて処理速度が遅い．そのため，比較的サイズが大きくなるデータ本体を共通鍵暗号によって暗号化し，その際に使用した鍵を公開鍵暗号によって共有する，ハイブリッド型暗号方式（ハイブリッド暗号）が用いられることがある．ハイブリッド暗号を用いることにより，処理時に時間のかかる公開鍵暗号の処理を最初に一度行い，共通鍵の共有を行うことで，それ以降は共通鍵暗号を用いてデータの送受信が行うことが可能になる．これにより，高速な動作を実現することが可能となる．

2.3　ID ベース暗号

公開鍵暗号方式の 1 つに ID ベース暗号（Identity-Based Encryption: IBE）が提案されている [17][18]．ID ベース暗号では個人を識別可能な ID を公開鍵として使用する．社員番号やメールアドレスなどがその例である．公開鍵と ID 情報を別々に持たなくてよいため鍵管理において負担が少ないといった特徴がある．ID ベース暗号では ID を公開鍵として暗号文を作成できるため，従来の公開鍵暗号では公開鍵の持ち主を証明するために公開鍵署名を用いる必要があったが，ID ベース暗号では不要となる．秘密鍵の発行は信頼できる鍵発行センタ（Key Generation Center: KGC）が行う．ユーザは KGC にて認証を行い，自身の ID に紐づいた秘密鍵を発行後に受け取る．これにより暗号文は，暗号化する際に使用した ID とマッチする秘密鍵を持つユーザのみが復号できる．

2.4　属性ベース暗号

近年，大学や企業などの組織は自組織でのサーバ運営コストを削減するため，オンラインストレージの利用へと移行する組織が増えている．しかし，オンラインストレージの運営は第三者が行っているため信用できない可能性が考えられる．そこで利用するクライアント側でデータの暗号化を行い，データと秘匿する必要がある．一般的には組織内でデータを共有する場合，個人間での共有よりも部署や役職ごとなどのグループ単位でのデータ

の共有を行う場合が多い．また複数の部署や役職間での共有がされる場合もあるため，暗号化を利用したきめ細やかなアクセス制御が求められるようになった．

そこで，ID ベース暗号で使用する ID を属性情報への置き換えに対応させた方式である属性ベース暗号（Attribute-Based Encryption: ABE）[19] が提案されている．属性ベース暗号では復号できるユーザの属性情報を指定して暗号化を行う．この際，AND や OR などを利用した属性情報の論理式を使って暗号化を行うことができる．これにより暗号化と合わせてきめ細やかなアクセス制御ができるため，複数人でのファイル共有に適している．属性ベース暗号には鍵ポリシー属性ベース暗号（Key-Policy Attribute-Based Encryption: KP-ABE）[20]，暗号文ポリシー属性ベース暗号（Ciphertext-Policy Attribute-Based Encryption: CP-ABE）[4] の 2 つの方式が存在する．KP-ABE は，暗号文にユーザの属性情報を埋め込み，復号鍵にアクセス権（復号できる属性情報）埋め込む方式である．この方式は動画配信などのコンテンツ配信の利用などに適しているとされている．CP-ABE は，暗号文にアクセス権を埋め込み，復号鍵に属性情報を埋め込む方式である．この方式は組織の秘密文書の共有などに適しているとされている．

CP-ABE[4] は所属や役職などの属性を公開鍵として利用する属性ベース暗号 [19] の一種である．属性の論理式で表現されたアクセス権（例：人事部 OR（総務部 AND 部長））を暗号文に埋め込むことで復号可能な人のグループを決定できる．受信者は鍵発行機関に自分の属性（例: 人事, 部長, ○○担当）が埋め込まれた秘密鍵を発行後に受け取り，秘密鍵に埋め込まれた属性集合が暗号文のアクセス権を満たすとき，暗号文を復号可能となる．CP-ABE の処理の概要を図 2.1 に示す．

CP-ABE は以下の 4 つのアルゴリズムから成る．

Setup(1^λ)　セキュリティパラメータ λ を入力しマスター公開鍵 PK とマスター秘密鍵 MK を生成し，出力する．

Encrypt(PK，M，A)　マスター公開鍵 PK と平文 M とアクセス権 A を入力すると，暗号文 CT を出力する．

KeyGen(MK，S)　マスター秘密鍵 MK と，秘密鍵を識別するための属性集合 S を入力すると，秘密鍵 SK を出力する．

Decrypt(PK，CT，SK)　マスター公開鍵 PK，秘密鍵 SK，暗号文 CT を入力すると，CT に埋め込まれたアクセス権 A にマッチする SK のみ平文 M を復号できる．

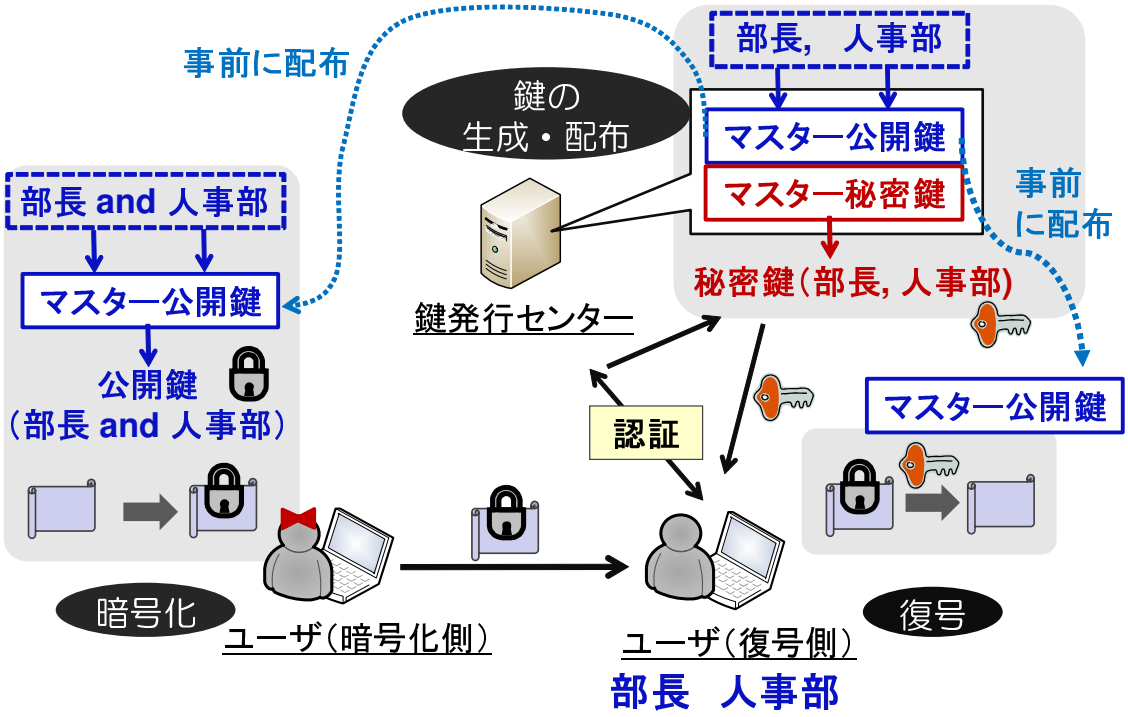

図 2.1　暗号文ポリシー属性ベース暗号の概要（許諾を得て文献 [1] より転載）

KGC は信頼できる機関であり，**Setup** で生成したマスター公開鍵とマスター秘密鍵を管理し，全ユーザにマスター公開鍵を配布する．ユーザ（暗号化側）は **Encrypt** でマスター公開鍵とアクセス権を利用して平文を暗号化する．属性に対応する秘密鍵は鍵発行機関が **KeyGen** でマスター秘密鍵と属性値を用いて発行する．ユーザ（復号側）は **Decrypt** でマスター公開鍵と秘密鍵を利用して暗号文を復号する．

ユーザが自身の秘密鍵を取得するとき，KGC はユーザの ID と属性の対応表を参照し，ユーザを認証した上でユーザの属性と紐付いた秘密鍵を (SSL/TLS を利用するなどして) 安全に配布する．ここで，属性が更新されない限りユーザの秘密鍵を変更する必要がないことに注意されたい．そのため，秘密鍵の配布の頻度は低いと考えられるため，鍵配布の処理時間は比較的大きい場合でも運用上問題にならないと考えられる．

CP-ABE は公開鍵暗号であるため柔軟な暗号化が可能であるが，AES などの共通鍵暗号と比べると低速である．これを解決するため，サイズが比較的大きいデータ本体は共通鍵暗号で暗号化し，それに用いる共通鍵（セッションキー）を CP-ABE で暗号化して保護する，ハイブリッド暗号が用いられることが多い．

2.4.1 ペアリング

属性ベース暗号の実現には暗号プリミティブとしてペアリングが用いられている．ペアリングは楕円曲線上の 2 点を入力とし，ある有限体の元を出力とする関数である．ペアリングの概要は以下のようになっている．p を λ ビット素数，$\mathbb{G}_1$，$\mathbb{G}_2$，$\mathbb{G}_T$ をそれぞれ位数 p を持つ巡回群とし，g_1，g_2 をそれぞれ $\mathbb{G}_1$，$\mathbb{G}_2$ の生成元とする．このとき任意の $a, b \in \mathbb{Z}_p$ に対し，$e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ が以下の双線形性と非退化性を満たすとき，この写像 e をペアリング写像と呼ぶ．

- **双線形性:** $e(g^a, g^b) = e(g, h)^{ab}$
- **非退化性:** $e(g, g) \neq 1_{\mathbb{G}_T}$

ここで $1_{\mathbb{G}_T}$ は $\mathbb{G}_T$ の単位元である．ペアリングには $\mathbb{G}_1 = \mathbb{G}_2$ である対称ペアリングと，$\mathbb{G}_1 \neq \mathbb{G}_2$ である非対称ペアリングがある．本研究で用いる MA-ABE の方式は対称ペアリングを用いて提案がされている．そのため，実装する際には対称ペアリング用の曲線である Type A Curve を用いた実装を行っている．実装方法の詳細に関しては，第 3.5.5 項にて説明を行う．

2.4.2 線形秘密分散法

秘密分散法 [21] は 1979 年に Shamir が提案した方式であり，秘密情報を複数の分散情報となるシェアに分散し，事前に設定された条件を満たすシェアを集めることにより元の秘密情報を復号できる方法である．Shamir が提案した閾値型秘密分散法は，シェアを生成する際に閾値をあらかじめ設定し，その閾値以上の個数のシェアを集めた場合に，秘密情報が復号できる方式となっている．属性ベース暗号のいくつかの方式では，アクセス権を表現する行列 A は秘密分散法の一種である，線形秘密分散法 (Linear Secret Sharing Scheme: LSSS) で作成する．LSSS では各シェアの論理式であるアクセスポリシーを事前に設定し，アクセスポリシーに合致するシェアを集めた場合に，秘密情報が復号できる方式となっている．アクセスポリシーはシェアを論理和（OR: $\vee$）や論理積（AND: $\wedge$）を用いて表現する．例えば A，B，C といったシェアがあった場合，A $\vee$ (B $\wedge$ C) といった表現で指定ができる．

本項では 2011 年に Lewko らが提案した LSSS [11] について概説する．この LSSS

は，秘密情報を復号する際に各行の加算の演算のみで実現可能な線形秘密分散法である．LSSS では秘密情報をシェアに分割する際に単純に分割等を行ってシェアを生成するわけではない．例えば AND 条件の場合，秘密情報 (s) に乱数 (r) を足したシェアと，乱数 ($-r$) のシェアに分割をする．この 2 つのシェアを足し合わせることにより，秘密情報 (s) が復号できる．この時それぞれのシェアは秘密情報 (s) に乱数 (r) が足されたものと，乱数 ($-r$) のみのシェアが生成されているのでシェア単体では秘密情報が復元できないことが分かる．OR 条件の場合はどちらのシェアを用いても秘密情報を復号できるようにするため，各シェアに同じ分散情報を割り当てることによりシェアを生成できる．シェアを生成する際にはアクセスポリシーからアクセス行列を生成し，各シェアをそれぞれ生成する．

まず，Lewko らの方式のアクセス行列生成方法の概要の例を図 2.2 に示す．この図の例ではアクセスポリシーは $E \wedge (((A \wedge B) \vee (C \wedge D)) \vee ((A \vee B) \wedge (C \vee D)))$ となっている．この方式ではまずグローバルカウンタ $c = 1$ とアクセスベクトル $v = 1$ を用意し，親ノードから子ノードへの分岐条件によってカウンタの値及びアクセスベクトルの値を変更するかどうかを決める．グローバルカウンタは親ノードに付与され，その子ノードのアクセスベクトルの要素数は c 個となる．以下はその決定方法である．ノードにグローバルカウンタとアクセスベクトルを決定する順番は，幅優先探索の順序のように，深さ順に一番上の親ノードから近いものから決定するようになっている．

親ノードが OR の場合: グローバルカウンタの値は変更せず，親ノードが持つアクセスベクトルの値を変更せず子ノードへと渡す．

親ノードが AND の場合: グローバルカウンタの値を直前の親ノードに付与したものから 1 上げたものを親ノードのグローバルカウンタの値として渡し，左の子ノードのアクセスベクトルは要素数 v は 0 を $c - 1$ 個と最後の要素を -1 とする．右の子ノードのアクセスベクトルは，親ノードのアクセスベクトルを最初の要素に入れ，その後ろに要素数が $c - 1$ 個になるように 0 で埋める．そして最後の要素に 1 を入れ，要素数 c のアクセスベクトルを生成する．

次に，作成されたアクセス行列から図 2.3 のように，秘密情報と乱数をそれぞれ用いて計算することにより，各シェアをそれぞれ生成する．その後，シェアを使って秘密情報を復号する場合，今回は $E \wedge (A \wedge B)$ の条件で復号する場合を例に説明を行う．図 2.3 より

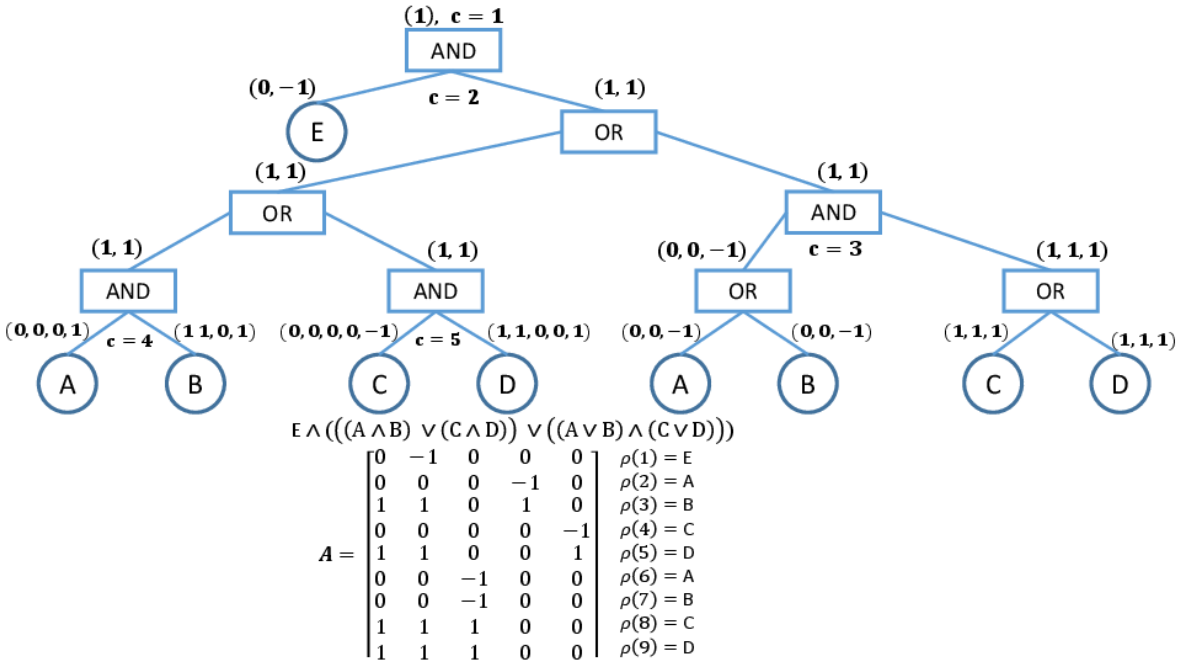

図 2.2　文献 [11] の線形秘密分散法の処理の概要

$$\begin{bmatrix} E \\ A \\ B \\ C \\ D \\ A \\ B \\ C \\ D \end{bmatrix} = \begin{bmatrix} 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 \\ 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & -1 \\ 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} s \\ r_1 \\ r_2 \\ r_3 \\ r_4 \end{bmatrix} = \begin{bmatrix} -r_1 \\ -r_3 \\ s + r_1 + r_3 \\ -r_4 \\ s + r_1 + r_4 \\ -r_2 \\ -r_2 \\ s + r_1 + r_2 \\ s + r_1 + r_2 \end{bmatrix}$$

s： 秘密情報, $r_{1\sim4}$： 乱数

図 2.3　アクセス行列を用いたシェア生成の計算方法

E，A，B のシェアの値はそれぞれ $-r_1$，$-r_3$，$s + r_1 + r_3$ であることが分かる．この 3 つの値を使用することにより，アクセスポリシーを満たしたシェアを集めたこととなるため，秘密情報の s が求められ，秘密情報の復号が行えたこととなる．このように，Lewko の方式では復号する際に使用するアクセスポリシーを満たすシェアの値を足し合わせることによって，復号を行うことができるようになっている．

2.5　ファイル共有システム

本節では，現在までに使用・提案されているファイル共有システムに関して概説する．はじめに，それぞれの既存のファイル共有システムに関して，そのシステムを用いることで実現できる目的と，使用する際に問題となる点に関して説明する．次に，既存のファイル共有システムの問題点を踏まえ，どのような要件を満たすファイル共有システムが必要となるのか考察を行う．

2.5.1　オンラインストレージを用いたファイル共有システム

近年オンラインストレージサービスの普及により，個人でのデータのバックアップのみならず，組織でのデータ共有等にもオンラインストレージが用いられる機会が増えている．オンラインストレージはファイル共有システムとして，手軽に使用できる方法の 1 つである．また，ファイルのアクセス制御を行うことで，保存してあるファイルにアクセスできるユーザを制御することが可能となっている．しかしながら，オンラインストレージサービスをそのまま使用する場合，サービス提供者によってディスクごとなどの暗号化はなされているが，これらはサービス提供者側によって提供されているものである．そのため，悪意のある管理者等が存在した場合に，データの流出等が発生する可能性が考えられる．そこで，これらのサービスを利用する場合はユーザ側で暗号化を行い，ファイルを保護する必要がある．

ユーザ自身が暗号化を行いファイル共有する方法として，VeraCrypt などのユーザ側でデータの暗号化を行い，その後オンラインストレージ上に保存するという方法がある．この方法では保存するファイルのセキュリティ管理をユーザ自身で行うため，適切に処理が行われている場合，オンラインストレージのサービス提供者側に悪意のある管理者が存在する場合などでも，ファイルの内容を保護することが可能である．しかしながら，この方法では共通鍵暗号などを用いてファイルの暗号化を行うため，ファイルを共有するグループごとに鍵を作成・配布しなくてはならない．そのため組織内などでのファイル共有を行う用途などでは運用コストが非常に大きくなってしまう．

2.5.2　CP-ABE を用いたファイル共有システム

従来のユーザによってデータの暗号化を行い，オンラインストレージ上で共有する方法では，鍵の管理に非常に大きな運用コストがかかってしまうことが問題であった．そこで，鍵管理のコストの問題を解消する方式として，公開鍵暗号の一種である属性ベース暗号を用いて，ファイル共有を行うシステムが提案されている [5][6][7]．ファイル共有システムは属性ベース暗号の最も期待される応用先と言えるため，CP-ABE を用いた様々なファイル共有システムに関して先行研究が行われている．これらのシステムでは，CP-ABE を用いることでオンラインストレージ上のファイルの閲覧権限を柔軟に制御でき，さらに CP-ABE によってデータの暗号化を行っている．ユーザは閲覧権限を有しているデータを復号する際に，使用する自身の属性が紐づけられた鍵のみでデータを復号することが可能となる．そのため，鍵管理のコストが非常に小さくなっている．しかし，これらのシステムではデータ自体の暗号化はされているが，ファイル名やディレクトリ名は保護されていない．ファイル名・ディレクトリ名はそのデータの要約など，重要な情報が含まれる場合もあるため，閲覧権限のないユーザに見られてしまうことは望ましくない．

CP-ABE を用いた従来のファイル共有サービスと異なり，コンテンツだけでなくファイル名・ディレクトリ名を含むディレクトリ構造全体を暗号化し，ファイル名・ディレクトリ名の秘匿，および編集権限の制御を行うシステムが先行研究として提案されている [2](図 2.4)．これにより，ファイル名・ディレクトリ名からデータの内容に関する情報が漏れることを防いでいる．このシステムでは，ファイル名やディレクトリ名をランダムな文字列に置き換え，その文字列と本来のファイル名・ディレクトリ名の対応を，ディレクトリごとのファイル (リストファイルと呼ぶ) で管理している．リストファイル内にてファイル名・ディレクトリ名を，閲覧が許可されているアクセス権ごとにまとめて CP-ABE で暗号化をすることで，高速に処理することを実現している．リストファイルは同じ属性を持つユーザが共有するため，同様の属性を持つユーザによってファイルの不正な書き換えや削除が起こることが考えられる．そこで，このシステムではディレクトリへのファイル・ディレクトリの追加処理を安全にするためにの閲覧権限と，編集権限を分けて制御を行う．これにはアップロードマネージャという登録専用のサーバを導入し，編集権限を集中管理する方法を用いている．アップロードマネージャでの処理も CP-ABE を用いた認証を利用することで，権限が無いユーザのファイルの登録も防いでいる．リストファイル

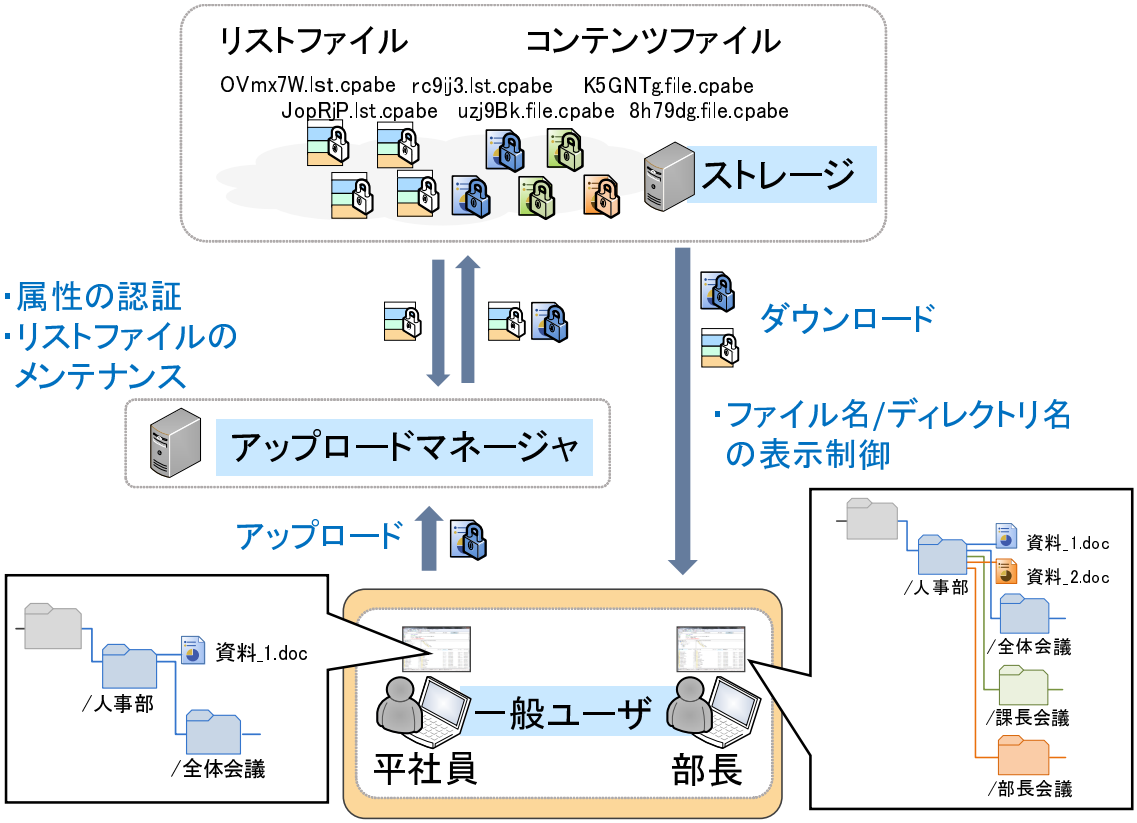

図 2.4　CP-ABE を用いたファイル共有システムの [2] の概要（許諾を得て文献 [1] より転載）

とアップロードマネージャのフォーマットや処理手順などに関しては，第 4.2.2 項にて詳細な記述を行う．

この先行研究で提案されているファイル共有システムには，CP-ABE が用いられている．そのため，CP-ABE を使用する際の制約となる単一組織での使用に限られてしまうシステムとなっている．しかしながら，実際のファイル共有システムの利用シーンを考えた場合，共同研究を複数組織間で行っている場合等を想定すると，研究データや論文のデータの共有など，共同研究に関するファイルを複数組織間で共有したい場合などが考えられる．先行研究では CP-ABE を用いており，制約上これらのユースケースに対応ができない．そこで，CP-ABE を用いた既存のファイル共有システムと同様の要件を満たしており，複数組織間で利用可能なファイル共有システムが必要となる．

2.5.3　複数組織間で利用可能なファイル共有システムの要件

第 2.5.2 項にて既存のファイル共有システムでは実現されていない，複数組織間で利用可能なファイル共有システムが必要となることが分かった．このシステムは既存のファイル共有システムで満たされている要件に加え，複数組織間で利用可能となることが求められる．このシステムで満たされるべき要件を以下に整理する．

- **要件 1：複数組織間で利用可能なファイル共有システムであること**
  共同研究等においてデータを共有する場合等において，研究室単位などの小規模で非常に数の多い組織等で共同で利用可能なシステムであること．
- **要件 2：共有元のユーザが指定した相手のみがファイルを編集・閲覧でき，他者にはファイルの内容がファイル名・ディレクトリ名も含めて漏洩しないこと**
  オンラインストレージなどを用いる場合でも，サービス提供者によるセキュリティ管理に頼らずユーザ自身でデータの暗号化を行うことにより，ファイルの安全性が確保されること．また，暗号化したデータの内容だけでなく，ファイル名・ディレクトリ名を含めた情報がサービス提供者を含めた対象のユーザ以外から秘匿できること．さらに，暗号化・復号する際の処理時間が著しく遅くないこと．
- **要件 3：ユーザの鍵管理が容易であること**
  使用する組織の数が多くなることが想定されるため，データを共有するユーザや組織の数が増えたとしても，要件 2 を満たすために必要な鍵管理のコストが大きくならないこと．
- **要件 4：ファイルを共有するユーザの指定を動的に変更できること**
  CP-ABE を用いた既存の共有システムを用いる場合と同様に，柔軟なアクセス制御を行えること．ユーザの属性変更時などの鍵失効や新規の鍵発行にも対応可能であること．

要件 1 は第 2.5.2 項で必要となることが示された，既存のファイル共有システムでは実現されていない，新たなファイル共有システムに求められる要件である．要件 2〜4 は，既存の CP-ABE を用いたファイル共有システムにてすでに満たされている要件であり，複数組織間で利用可能なファイル共有システムでも同様に満たされているべき要件である．新たに複数組織間で利用可能なファイル共有システムを開発するにあたって，以上の 4 つ

の要件を満たすことが必要となる．

2.6　結言

本章では，本研究に関連する先行研究と関連技術について述べた．まず，公開鍵暗号の概要について説明し，その後，公開鍵暗号の一種である ID ベース暗号について概説した．次に，先行研究で使用されている属性ベース暗号に関して説明した．その後，既存の CP-ABE を用いたファイル共有システムについて概説した．最後に，それぞれのファイル共有システムの特性を踏まえ，複数組織間で利用可能なファイル共有システムが必要であることを示した．そして，複数組織間で利用可能なファイル共有システムに求められる要件について整理を行い，その概要を述べた．

第 3 章

研究データ向けファイル共有システムの提案・考察

3.1　緒言

本章では, 第 2.5 節にて必要性が示された複数組織間で利用可能なファイル共有システムの提案・考察を行う．一般的に，従来のファイル共有システムは大学単位などの比較的規模の大きな組織ごとに管理がなされている．しかしながら，実際のユースケースでは研究室単位や共同研究における研究グループごとなど，小規模な単位の組織間でデータにアクセスできるグループを作成したい場合がある．たとえば，複数の研究室間で共同研究を行っていた場合，大学単位で管理されている従来のファイル共有システムを使ったとすると，大学側の管理者に研究データを見られてしまう可能性がある．そのため，研究グループ内にのみ公開したいデータを扱う場合には，このような使用方法は好ましくない．特に学外の研究機関と共同研究している場合，特に研究室単位で秘密保持契約 (Non-Disclosure Agreement: NDA) を締結しているような場合において，大学単位での KGC を持つシステムでは利用に適さない．そこで本章では，第 2.5.3 項にて挙げられた要件を満たす，研究データ向けファイル共有システムを提案する．そのために，第 2.5.3 項にて挙げられた要件 1 以外をすでに満たしている，既存の CP-ABE を用いたファイル共有システム [2] を複数組織で利用できる形に対応させることを検討する．従来のファイル共有システムにおいて単一組織でしか使えなくなっていた原因である CP-ABE を使用する代わりに，MA-ABE を用いることにより，研究データ向けの複数組織間で利用可能なファイル共有システムを実現する方法を提案する．

本章ではまず，要件 1〜4 に沿った形で，提案するファイル共有システムに適した MA-ABE を選定する．次に，MA-ABE の各アルゴリズムや通信の処理時間の計測・評価を行うことで，提案方式に用いる MA-ABE の定量的性能評価を行う．これにより，提案するファイル共有システムが実用可能であることを示す．さらに，提案システムを実際に運用する場合，ユーザの人事異動，卒業，退学などによる属性情報の変更が生じることが想定される．これらに対応するための鍵失効に関する問題，属性管理についての考察を行う．以上の結果より，要件 1〜4 を満たす研究データ向けファイル共有システムの実用性を明確化する．

3.2　複数組織対応属性ベース暗号 (MA-ABE)

単一組織のみでの使用が想定されている従来の属性ベース暗号 [4][22] では，該当する KGC をすべての組織で信頼し共有しなければならず，KGC の分散管理は困難である．このような場面でも属性ベース暗号を複数組織で利用できるように，KGC を組織ごとに用意して，複数組織で連携して利用可能な属性ベース暗号 (Multi-Authority Attribute-Based Encryption: MA-ABE) が提案されている（図 3.1）．MA-ABE には大きく分けて，各組織の KGC を中央機関によって管理する方式 [8] と，中央機関を必要とせず KGC が複数存在可能な方式 [9][10][11][12][13] が存在する．これらの方式は，複数組織で利用する場合などの現実的な属性管理ができるため，従来の KGC が 1 つのみの属性ベース暗号と比べ優れている．

中央機関を必要とせず複数の KGC が存在可能な属性ベース暗号では，ユーザ同士の結託攻撃に対する耐性を実現する必要がある．そこで，これらの方式では，MA-ABE を利用する全組織の全ユーザにおいて，ユーザごとに固有の識別子 GID を決め，KGC に依らず GID を秘密鍵生成の演算に含めることで結託耐性を与えている．たとえば，「A 大学教員かつ B 大学客員研究員」のユーザ向けの暗号文を解読するために，「A 大学教員」と「B 大学客員研究員」がそれぞれ秘密鍵を提供し合ったとしても，それぞれの秘密鍵の GID が異なることから結合ができないため解読を防ぐことができる．

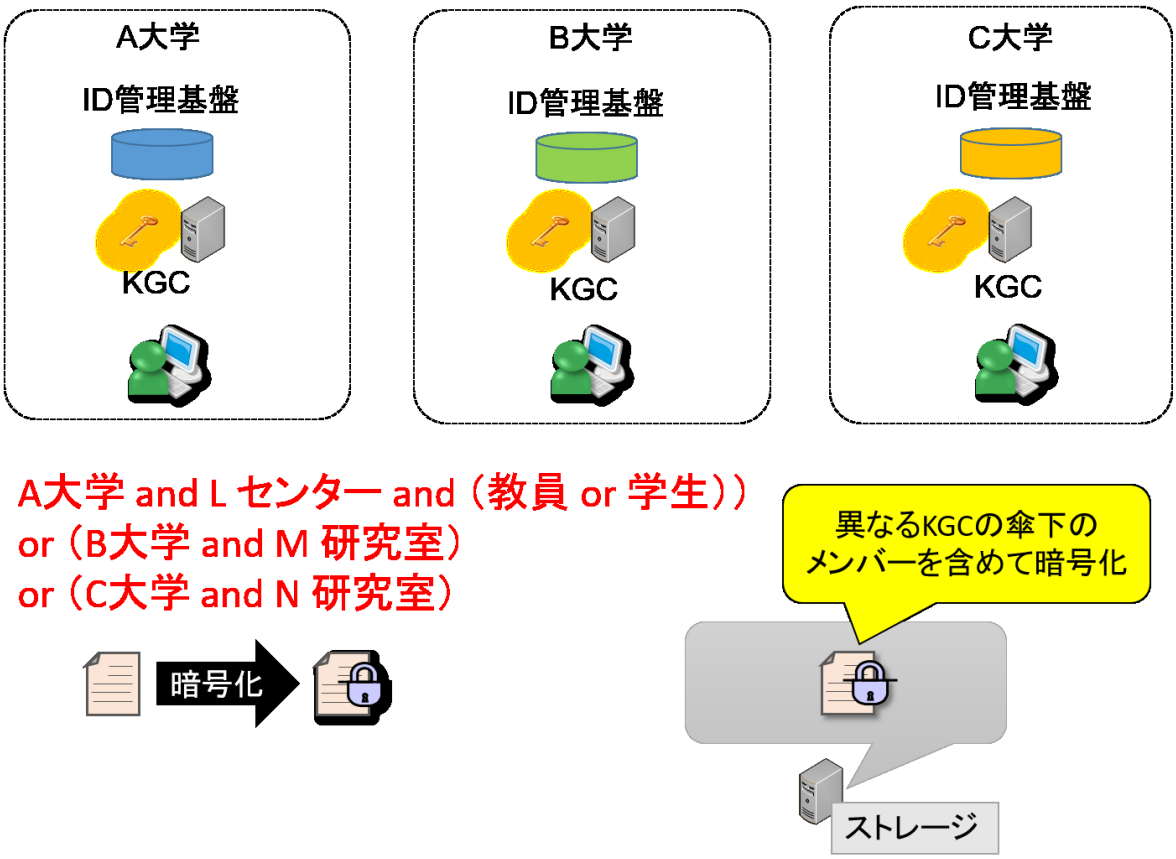

図 3.1　複数組織での KGC 管理の概要（許諾を得て文献 [1] より転載）

3.3　MA-ABE を用いた複数組織で利用可能なファイル共有システムの提案

本研究では，従来のファイル共有システムにおける鍵管理の単位を，学部単位や研究室単位などの比較的規模の小さく数の多い組織にて行うことを可能とする．そこで，先行研究にて提案されている CP-ABE を用いたファイル共有システム [2] において，MA-ABE を用いることにより複数組織対応化を実現させる方法を提案する．これにより，提案するファイル共有システムにおいては，研究グループの責任者が秘密保持契約の代表者となるような場合にも対応し，細分化した KGC の設置と管理を可能にする．秘密保持契約の代表者と KGC の管理者を一致させることで，権限を持たない者によるデータの覗き見を防止できる．提案システムはファイル共有において，オンラインストレージ単体で使用する場合よりも安全性は高くなるが，処理時間がオンラインストレージ単体で使用するときに

比べ著しく遅くなる場合，ユーザの利便性が損なわれることが想定される．そのため，提案システムの各処理に関しては，オンラインストレージ単体でのファイル共有時と比べて，処理時間に差異が少ないことが求められる．また，オンラインストレージ単体で利用する場合の利点として，ファイルを共有するユーザの属性変更等におけるアクセス制御の柔軟さが考えられる．そのため，提案システムではアクセス制御は MA-ABE の機能の一部であるが，ユーザの属性変更時などにおける鍵失効機能も実現が必要となる．第 3.1 節で挙げた要件を満たすために使用する MA-ABE に関しての比較・検討を第 3.4 節，次に，処理時間の計測・評価を第 3.5.5 項にて行う．その後，提案ファイル共有システムを使用する際に必要となると考えられる．鍵失効機能の方法に関する考察を第 3.6 節にて行う．

本研究では，KGC を動作させるサーバの運用・管理は KGC の管理者と同一組織が行うことを想定している．これは KGC が鍵生成を行うための秘密鍵 SK を保存しているサーバを安全に管理する必要があるからである．しかしながら，KGC の管理を研究室等の秘密保持契約の代表者単位まで細分化した結果，大学組織等で運用される場合と異なり，研究室単位等で管理された KGC は冗長化されていない場合も想定される．そのような場合に運用が現実的であるかについて，故障時の影響の大きさの視点で検討する．KGC が動作しているサーバが何らかの原因で故障した場合，再起動後に主要機能である鍵生成（KeyGen）を再開するためには KGC が有している秘密鍵 SK があれば良い．つまり，KGC の秘密鍵 SK を外部デバイス等にバックアップし安全に保管をしておけば，代替のサーバを確保できた時点で復旧が可能となる．また KGC にアクセスするタイミングは，ユーザが最初に自身の属性に対応する秘密鍵の発行を依頼する際の一度のみであり，暗号化や復号などの処理毎など頻繁に KGC へのアクセスが発生することはない．そのため，停電や故障等で一時的に KGC へのアクセスができなくなったとしても，影響を受けるのは鍵の発行をこれから新規に行うユーザのみであり限定的である．したがって，サーバがダウンした場合に影響が大きいストレージサービス等と比べ，KGC の冗長化がされていなかったとしても，障害発生時の影響は小さいと考えられるため，研究室等の小規模組織での運用も現実的であると考えられる．なお，KGC の運用負荷の軽減のためにクラウドサーバで KGC を管理することも考えられるが，重要な KGC を外部に委託することになるため，慎重な検討が必要となる．その詳細は第 3.7.2 項で考察する．

3.4　提案システムに適する MA-ABE の調査・選定

複数組織で利用可能なファイル共有システムを提案するにあたり，MA-ABE を用いてファイル共有システムを構築する方法を示す．そこで，本節では MA-ABE の複数の方式に関して，今回提案するファイル共有システムに一番適している方式を調査・選定する．

3.4.1　MA-ABE の方式比較

現在，MA-ABE は様々な方式が提案されている [8][9][10][11] [12][13]．これらの方式は 3.2 節で説明した通り，方式自体に結託耐性があるため，AND 表現のときに問題になった結託攻撃に対して安全性を有している．表 3.1 は，調査した方式 [8][9][10] [11][12][13] を以下の条件で分類したものである．

- **中央機関は必要か？**
  KGC を分散管理する方式として，中央機関を必要とするかどうか．
- **楕円曲線の位数**
  対称ペアリング暗号に用いられる楕円曲線の位数．
- **安全性**
  その方式で証明されている安全性証明．
- **公開パラメータのサイズは属性数に比例せず一定か？**
  公開パラメータのサイズが属性数に比例することなく一定かどうか．

今回提案する MA-ABE を用いた複数組織で利用可能なファイル共有システムには，全部で 4 つの満たすべき要件が存在する．それらの要件を満たすためには，提案システムに用いる MA-ABE は以下の 2 点を満たしている必要がある．

- **中央機関が不必要**
  中央機関を用いて KGC を分散管理する方式は，中央機関に対応する KGC が他の全ての KGC の秘密鍵を生成できてしまう．そのため，中央機関が存在する場合は CP-ABE を用いた既存の共有システムと同様，中央機関の管理に関する問題が発生する．したがって，提案システムの要件 1 を満たすために，中央機関が不必要な方式 (De-centralized MA-ABE) である必要がある．

表 3.1　複数の KGC が存在可能な属性ベース暗号の分類（許諾を得て文献 [1] より転載）

	中央機関は必要か？	楕円曲線の位数	安全性	公開パラメータのサイズは属性数に比例せず一定か？
Chase の方式 [8]	必要	Prime	Selective	No
Lewko らの方式 [11]	不必要	Composite	Adaptive	No
岡本らの方式 [12]	不必要	Prime	Adaptive	No
Datta らの方式 [13]	不必要	（格子ベースの方式のため楕円曲線は用いない）	Selective	No
Lewko の方式 [10]	不必要	Prime	Adaptive	No
Rouselakis らの方式 [9]	不必要	Prime	Selective	Yes

- **楕円曲線の位数に prime order を使用した方式**

  MA-ABE は対称ペアリング暗号を用いて実現されている．使用する楕円曲線に合成数位数 (composite order) 楕円曲線を用いる方式は，素数位数 (prime order) 楕円曲線を用いる方式よりも計算量が多く，パラメータサイズも大きくなる [23]．そのため，提案システムの要件 2 にて述べている，暗号化・復号にかかる処理時間が遅くならないという要件を満たすために，合成数位数楕円曲線を用いた方式は採用しない．

表 3.1 より，文献 [8] の方式は中央機関を必要とするため採用せず，文献 [11] の方式は楕円曲線の位数に composite order を使用するため採用しないこととした．また，文献 [13] の方式は Selective 安全性での提案方式であるが，公開パラメータのサイズが属性数に比例して大きくなってしまうため，この 2 つの観点で比較した際に他の方式の方が優れているため採用しない．

Lewko の方式 [10] と岡本らの方式 [12] は，公開パラメータのサイズが各 KGC の管理する属性数に比例して大きくなるといった方式である．表中の方式のうち，公開パラメータのサイズ以外の点では Lewko の方式と岡本らの方式の 2 つの方式が優れた性質を達成している．この 2 つの方式を比較すると，KGC の公開パラメータに用いる要素のサイズを同条件とした場合に，Lewko の方式の方が公開パラメータのサイズが小さくなる．そこで本節では，上記の全ての条件を満たす方式の中から Lewko の方式 [10] をシステムに用いるための検討を行うこととする．また，属性数に関係なく公開パラメータのサイズが一定である方式のうち，実装上比較的シンプルな構成で実装可能な方式である Rouselakis らの方式 [9] においても更に詳しく比較・評価をすることにした．この二つの方式は，証明に用いられている安全性にも違いがある．そこで公開パラメータのサイズの違い，安全

性の違いを含めた比較を第 3.4.2 項にて行う．

今回比較に用いている Rouselakis らの方式 [9] をベースに traceability を実現した方式 [24] の提案もされている．しかしながら，安全性証明などの条件は Rouselakis らの方式と同様のものであり今回の比較対象には含めていない．今後，提案システムにおいて traceability を使用する必要が出てきた場合には，別途検討が必要となる可能性も考えられる．

3.4.2 MA-ABE の安全性の比較

本節では，Lewko の方式 [10] と Rouselakis らの方式 [9] で証明されている安全性証明の違いによる比較を行う．今回の 2 つの方式の安全性に関しては，Lewko の方式は Rouselakis らの方式より強い Adaptive 安全性を実現している．対して，Rouselakis らの方式では Selective 安全性しか達成されていない．ここで，Selective 安全性とは，Setup よりも前の段階で攻撃者が攻撃対象となるアクセス権を，チャレンジャに発行することが強いられているモデルでの安全性となる．一方で，Adaptive 安全性とは，鍵クエリを行った後に攻撃者が攻撃対象となるアクセス権をチャレンジャに発行するモデルでの安全性となる．

Lewko の方式では, Adaptive 安全性を実現するために DPVS(Dual Pairing Vector Spaces) を用いているので，公開パラメータのサイズは一定ではなくなる．Rouselakis らの方式では，Selective 安全性での証明となっているものの，公開パラメータのサイズは一定となる．今回の提案システムにおいて，より安全性の高い MA-ABE を用いるのが好ましいことは明白である．しかし，これらの公開パラメータが，提案システムにて実際に利用する場合に，どの程度影響が出るのか考察を行う必要がある．以上の考察を第 3.5 節にて行う．

なお近年，Venemar と Alpar により MA-ABE に対する攻撃方法 [25] が提案されている．しかし，この攻撃は中央機関を必要とするタイプの MA-ABE に対する汎用的な攻撃である．今回比較する Lewko の方式 [10] や Rouselakis らの方式 [9] に対する攻撃は文献 [25] の中で提案されておらず，また，個別の攻撃方法も知られていない方式のため，現時点でこの方式は安全である．

3.5　MA-ABE を用いたファイル共有システムの評価

本節では，MA-ABE を用いたファイル共有システムを実現するに当たって必要と考えられる MA-ABE の実装・評価及び，システムの通信時間を含めた処理時間の計測・評価を行う．まず第 3.5.1 項では，第 3.4 節で調査した MA-ABE の方式である Lewko の方式と Rouselakis らの方式のアルゴリズムを示す．第 3.5.4 項では，それぞれの方式の公開パラメータの比較を行う．第 3.5.5 項では，それぞれの方式のアルゴリズムごとの処理時間の計測を行う．これにより Rouselakis らの方式が方式自体は実利用に向いているが処理時間に問題が無いことを調査する．この時，Rouselakis らの方式が実利用不可能なほど処理時間が遅かった場合，次点で実利用するのに適していると考えられる Lewko の方式を比較対象として比較を行う．第 3.5.6 項では，Rouselakis らの方式をファイル共有システムに使用すると想定した場合，アクセス権として定義する属性の数の変化によって，暗号化および復号にかかる処理時間が変化すると考えられる．そこで，暗号化・復号それぞれ属性数を変化させての処理時間の計測・評価を行う．第 3.5.6 項では，実際のシステムとして使用する際にデータを保存・閲覧する際に生じる，ユーザとサーバ間で通信を含めた暗号化・復号の処理時間の計測を行う．この計測では通信時間を含めた処理時間の計測を行うことにより，実際のシステムを利用する際に一番多く発生する処理であるデータの保存・閲覧が，現実的な時間で動作することの確認を行う．

3.5.1　複数組織対応属性ベース暗号のアルゴリズム

本節では第 3.4 節で選定した, Lewko の方式 [10] と Rouselakis らの方式 [9] の詳細なアルゴリズムを示す.

3.5.2　Lewko の方式のアルゴリズム

暗号化の際のアクセス権に KGC の識別番号および属性を紐づけるベクトル ρ と DPVS を用いて複数の組織（KGC）が連携した暗号化を実現する．Lewko の方式のシンタックスを以下に示す.

定義 1　(Lewko の方式 [10])

Lewko のアルゴリズムは以下の 5 つのアルゴリズムで構成される.

Global Setup: Global Setup はセキュリティパラメータ λ を入力とし，グローバルパラメータ GP を出力する．

Authority Setup: Authority Setup は GP を入力とし，公開パラメータ PK と秘密鍵 SK を出力する．

Encrypt: Encrypt は GP，データ M，$\ell \times n$ 行のアクセス行列 (A, ρ)，PK を入力とし，暗号文 CT を出力する．ここで A は属性の論理式に合致したときに復号されるように LSSS を用いて作成された行列であり，ρ は A に対応する KGC および属性を紐づけるためのベクトルである．

KeyGen: KeyGen は GP，ユーザの識別子 i，ユーザ固有の識別子である GID，SK を入力とし，復号するユーザの秘密鍵 $K_{i,\mathrm{GID}}$ を出力する．

Decrypt: Decrypt は GP，CT，$K_{i,\mathrm{GID}}$ を入力とし，データ M を出力する．

正当性として, Global Setup(1^λ), Authority Setup(GP), CT $\leftarrow$ Encrypt(GP, M, (A, ρ), PK), $(K_{i,\mathrm{GID}}) \leftarrow$ KeyGen(GP, i, GID, SK) に対し，$M = \mathrm{Decrypt}(\mathrm{GP}, \mathrm{CT}, K_{i,\mathrm{GID}})$ が成り立つことを要求する．

Lewko の方式で用いている対称ペアリングを次のように定義する．$\mathbb{G}$ と $\mathbb{G}_T$ を λ ビットの素数位数 p を持つ群とし，$g \in \mathbb{G}$ を生成元とする．e を双線形写像 $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ とし，任意の $a, b \in \mathbb{Z}_p$ に対し，$e(g^a, g^b) = e(g, h)^{ab}$ が成り立つ双線形性，$e(g, g) \neq 1_{\mathbb{G}_T}$ が成り立つ非退化性をみたすとする．ここで $1_{\mathbb{G}_T}$ は $\mathbb{G}_T$ の単位元である．このとき，Lewko の方式のアルゴリズムの詳細は以下のように与えられる．

Global Setup (1^λ): $\mathbb{G}, \mathbb{G}_T$ を λ ビットの素数位数 p を持つ群とする．$g \in \mathbb{G}$ を生成元とし，$e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ を双線形写像とする．さらに，H: $\{0,1\}^* \to \mathbb{G}_T$ をハッシュ関数と定義する．GP = $\{\mathbb{G}, p, g, H\}$ を出力する．

Authority Setup (GP): KGCi は 12 次正方行列 $B \in \mathbb{Z}_p^{12\times 12}$ をランダムに生成し，正規直交基底の組 $(\mathbb{B} = (\vec{b}_1, \ldots, \vec{b}_{12}), \mathbb{B}^* = (\vec{b}_1^*, \ldots, \vec{b}_{12}^*)) \in \mathbb{Z}_p^{12\times 12}$ を生成する．さらに，KGCi は 2 つの一様乱数 ${\alpha_i}^1, {\alpha_i}^2 \in \mathbb{Z}_p$ を選択する．KGCi は公開パラメータ PK = $\{e(g,g)^{{\alpha_i}^1}, e(g,g)^{{\alpha_i}^2}, g^{\vec{b}_{1,i}}, g^{\vec{b}_{2,i}}, g^{\vec{b}_{3,i}}, g^{\vec{b}_{4,i}}\}$ と秘密鍵 SK = $\{g^{{\alpha_i}^1 \vec{b}_{1,i}^*}, \vec{b}_{1,i}^*, \vec{b}_{2,i}^*, \vec{b}_{3,i}^*, g^{{\alpha_i}^2 \vec{b}_{3,i}^*}, \vec{b}_{4,i}^*\}$ を出力する．

Encrypt (GP, M, (A, ρ), PK): 暗号化するユーザは一様乱数 $s \in \mathbb{Z}_p$ を選択する．アクセス権が定義されている $\ell \times n$ 行のアクセス行列 (A, ρ) に関して，要素数 n の

ベクトル $v, w^1, w^2 \in Z_p^n$ をランダムに選択する．ただし，v の 1 つ目の要素を s とし，w^1, w^2 の 1 つ目の要素を 0 とする．さらに，$j \in [1, \ell]$ に対し r_j^1, r_j^2 をランダムに選択する．$C = Me(g,g)^s, D_j = e(g,g)^{A_j \cdot v} e(g,g)^{\alpha^1_{\rho(j)} r_j^1} e(g,g)^{\alpha^2_{\rho(j)} r_j^2}$, $C_j = g^{r_j^1 \vec{b}_{1,\rho(j)}} + (r_j^1 + A_j \cdot w^1)\vec{b}_{2,\rho(j)} + r_j^2 \vec{b}_{3,\rho(j)} + (r_j^2 + A_j \cdot w^2)\vec{b}_{4,\rho(j)}$ を出力する．暗号文 CT は (A, ρ) と $C, \{D_j\}, \{C_j\}$ によって構成される．

KeyGen (GP, i, GID, SK)：　KGCi は $H(\mathrm{GID}) = (H^1_{\mathrm{GID}}, H^2_{\mathrm{GID}}) \in \mathbb{G}$ を計算する．復号するユーザの秘密鍵 $K_{i,\mathrm{GID}} = g^{\alpha_i^1 \vec{b}^*_{1,i}} g^{\alpha_i^2 \vec{b}^*_{3,i}} (H^1_{\mathrm{GID}})^{\vec{b}^*_{1,i} - \vec{b}^*_{2,i}} (H^2_{\mathrm{GID}})^{\vec{b}^*_{3,i} - \vec{b}^*_{4,i}}$ を出力する．

Decrypt (GP, CT, $\{K_{i,\mathrm{GID}}\}$)：　アクセス権が定義されている $\ell \times n$ 行のアクセス行列 (A, ρ) 下で復号するユーザは $\sum_{j=1}^{l} \omega_j A_j = (1, 0, \ldots, 0)$ になるような $\omega_1, \ldots, \omega_j \in \mathbb{Z}_p$ を選ぶ．ただし，$\rho(j)$ が秘密鍵を持っている復号するユーザの属性であった場合，$\omega_j \neq 0$ とする．復号するユーザは $F_j = D_j / e_{12}(K_{\rho(j),\mathrm{GID}}, C_j)$ を計算し，$M = C / \prod_{j=1}^{\ell} F_j^{\omega_j}$ を出力する．

ここで $e_{12}(\cdot,\cdot)$ は DPVS の計算である．ペアリング演算 $e(\cdot,\cdot)$ は 2 つの曲線の変数が入力される演算であったが，DPVS はそれをベクトルに拡張したものである．$e_{12}(\cdot,\cdot)$ は 12 個の要素を持つベクトルが 2 つ入力され，各ベクトルの要素間でペアリング演算を行い，計算結果を全て乗算する演算となる．たとえば，$\vec{v} = (v_1, \ldots, v_{12}), \vec{w} = (w_1, \ldots, w_{12}) \in \mathbb{Z}_p^{12}, g \in \mathbb{G}$ のときは $e_{12}(g^{\vec{v}}, g^{\vec{w}}) = \prod_{i=1}^{12} e(g^{v_i}, g^{w_i}) = e(g,g)^{\vec{v} \cdot \vec{w}}$ のように演算する．

正当性は以下の式より証明できる．

$e_{12}(K_{\rho(j),\mathrm{GID}}, C_j) = e(g,g)^{\alpha_{\rho(j)^1} r_j^1} e(g,g)^{\alpha_{\rho(j)^2} r_j^2} e(g, H^1_{\mathrm{GID}})^{-A_j \cdot w^1} e(g, H^2_{\mathrm{GID}})^{-A_j \cdot w^2}$
したがって，$F_j = e(g,g)^{A_j \cdot v} e(g, H^1_{\mathrm{GID}})^{A_j \cdot w^1} e(g, H^2_{\mathrm{GID}})^{A_j \cdot w^2}$ なので
$\prod_{j=1}^{\ell} F_j^{\omega_j} = e(g,g)^{\sum_{j=1}^{\ell} \omega_j A_j v} = e(g,g)^s$ ($\sum_{j=1}^{l} \omega_j A_j$ は法 p で w^1 と w^2 が直交となるため成り立つ).

Lewko の方式では，KeyGen において KGC 番号である i のみを入力として使用しており，ユーザの属性情報を使用していない．これは 1 つの KGC で扱うことができる属性が一種類であることを示している．

3.5.3 Rouselakis らの方式のアルゴリズム

まず初めに Rouselakis らの方式 [9] のシンタックスを以下に示す.

定義 2 Rouselakis らのアルゴリズムは以下の 5 つのアルゴリズムで構成される.

Global Setup: Global Setup はセキュリティパラメータ λ を入力とし，グローバルパラメータ GP を出力する.

Authority Setup: Authority Setup は GP と KGC 番号 θ を入力とし，公開パラメータ PK_θ と秘密鍵 SK_θ を出力する.

KeyGen: KeyGen は GP，KGC_θ 内のユーザの属性情報 u，ユーザ固有の識別子である GID，SK_θ を入力とし，復号するユーザの秘密鍵 $\mathrm{SK}_{\mathrm{GID},u}$ を出力する.

Encrypt: Encrypt は GP，データ M，アクセス構造 $\mathbb{A} = (A, \delta)$，$\ell \times n$ のアクセス行列が A，δ が A の各行と属性を結びつける関数となる．なお，各属性に対応する公開鍵の集合 $\{\mathrm{PK}_\theta\}$ を入力とし，暗号文 CT を出力する．ここで A は LSSS を用いて作成された行列である．つまり，A の各行を $A_i (i = 1, \ldots, \ell)$，アクセス権を満たす属性の集合に対応したラベルの集合を $I \subseteq [\ell]$ としたとき，$\sum_{i \in I} c_i A_i = (1, \ldots, 0)$ となるような $\{c_i\}_{i \in I} (c_i \in \mathbb{Z}_p)$ が存在する.

Decrypt: Decrypt は GP，CT，$\{\mathrm{SK}_{\mathrm{GID},u}\}$ を入力とする．ここで，$\{\mathrm{SK}_{\mathrm{GID},u}\}$ に関連するユーザ GID の属性の集合 $\Gamma_{\mathrm{GID}} := \{(\mathrm{GID}, u)\}$ が CT に付与されたアクセス権を満たすなら，データ M を出力する．そうでないならば，$\perp$ を出力する.

正当性として，GP $\leftarrow$ Global Setup(1^λ)，$(\mathrm{PK}_\theta, \mathrm{SK}_\theta) \leftarrow$ Authority Setup(GP, θ)，$\mathrm{SK}_{\mathrm{GID},u} \leftarrow$KeyGen(GID, θ, u, SK_θ, GP)，CT$\leftarrow$ Encrypt $(M, (A, \delta), \{\mathrm{PK}_\theta\}, \mathrm{GP})$ に対し，$\{\mathrm{SK}_{\mathrm{GID},u}\}$ に関連するユーザ GID の属性の集合 Γ_{GID} が CT に付与されたアクセス権を満たすなら，M = Decrypt $(\mathrm{CT}, \{\mathrm{SK}_{\mathrm{GID},u}\}, \mathrm{GP})$ が成り立つことを要求する.

Rouselakis らの方式で用いている対称ペアリングを次のように定義する．対称ペアリング群 $\mathsf{param} = (p, \mathbb{G}, \mathbb{G}_T, g, e)$ はそれぞれ，ビット長 λ の素数位数 p の乗法的巡回群 $\mathbb{G}, \mathbb{G}_T$，生成元 $g \in \mathbb{G}$，多項式時間で計算可能な非退化性を有する双線形写像 $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ から成る．セキュリティパラメータ λ を入力に取り，対称ペアリング群 param を出力するアルゴリズムを $\mathcal{G}_{\mathsf{SPG}}(1^\lambda)$ とする．このとき，Rouselakis らの方式のア

ルゴリズムの詳細は以下のように与えられる．

Global Setup (1^λ)**:**　$\mathsf{param} \leftarrow \mathcal{G}_{\mathsf{SPG}}(1^\lambda)$ を実行する．また，U, U_Θ はそれぞれ属性，KGC 番号である θ の母集団，T は属性からその属性が属している KGC へとマッピングする関数と定義し，さらに，H は GID を，F は文字列をそれぞれ $\mathbb{G}$ の要素へとマッピングするハッシュ関数とする．GP = $\{\mathsf{param}, H, F, U, U_\Theta, \mathsf{T}\}$ を出力する．

Authority Setup (GP, θ):　KGCθ は，2 つの一様乱数 $\alpha_\theta, y_\theta \xleftarrow{R} \mathbb{Z}_p$ を選択する．KGCθ は公開パラメータ PK = $\{e(g,g)^{\alpha_\theta}, g^{y_\theta}\}$ と秘密鍵 SK = $\{\alpha_\theta, y_\theta\}$ を出力する．

KeyGen (GID, θ, u, SK_θ, GP):　KGCθ は $t \xleftarrow{R} \mathbb{Z}_p$ を選択する．そして，ユーザの秘密鍵 $\mathrm{SK}_{\mathrm{GID},u} = \{\mathrm{K}_{\mathrm{GID},u} = g^{\alpha_\theta} H(\mathrm{GID})^{y_\theta} F(u)^t, \mathrm{K}'_{\mathrm{GID},u} = g^t\}$ を出力する．

Encrypt ($M, (A, \delta), \{\mathrm{PK}_\theta\}, \mathrm{GP}$)**:**　暗号化するユーザは一様乱数 $z, v_2, \ldots, v_n, w_2, \ldots, w_n \xleftarrow{R} \mathbb{Z}_p$ を選択し，ベクトル $v = (z, v_2, \ldots, v_n)^\top$，$w = (0, w_2, \ldots, w_n)^\top$ を生成する．そして A_x と v の内積を計算し，$\lambda_x = \langle A_x, v\rangle$ とする．同様に $\omega_x = \langle A_x, w\rangle$ も計算する．$t_x \xleftarrow{R} \mathbb{Z}_p$ を生成して，$C_0 = Me(g,g)^z, \{C_{1,x} = e(g,g)^{\lambda_x} e(g,g)^{\alpha_{\rho(x)} t_x}, C_{2,x} = g^{-t_x}, C_{3,x} = g^{y_{\rho(x)} t_x} g^{\omega_x}, C_{4,x} = F(\delta(x))^{t_x}\}_{x \in [\ell]}$ を出力する．なお，$\rho : [\ell] \to U_\Theta$，つまり $\rho(\cdot) = T(\delta(\cdot))$ とする．

Decrypt ($\mathrm{CT}, \{\mathrm{SK}_{\mathrm{GID},u}\}, \mathrm{GP}$)**:**　復号するユーザは自身の復号用の鍵 $\{\mathrm{SK}_{\mathrm{GID},u}\}$ を使用し，暗号文 CT から平文 M を計算し出力する．ここで，A の各行を A_x，アクセス権を満たす属性の集合に対応したラベルの集合を $I \subseteq [\ell]$ としたとき，$\sum_{x \in I} c_x A_x = (1, \ldots, 0)$ とする．

正当性は以下の式より証明できる．

$$
\begin{aligned}
C_{1,x} \cdot e(\mathrm{K}_{GID,\delta(x)}, C_{2,x}) \cdot e(\mathrm{H}(\mathrm{GID}), C_{3,x}) \\
\cdot e(K'_{\mathrm{GID},\delta(x)}, C_{4,x}) = e(g,g)^{\lambda_x} e(H(\mathrm{GID}), g)^{\omega_x}, \\
\prod_{x \in I} \left(e(g,g)^{\lambda_x} e(H(\mathrm{GID}), g)^{\omega_x} \right)^{c_x} = e(g,g)^z, \\
M = C_0 / e(g,g)^z.
\end{aligned}
$$

3.5.4　公開パラメータのサイズ比較

本節では，Lewko の方式と Rouselakis らの方式をファイル共有システムに使用する場合の，公開パラメータの具体的なサイズの比較を行う．第 3.2 節で示されているアルゴ

リズムから，Rouselakis らの方式ではユーザの復号用の鍵を生成する KeyGen において，ユーザの属性情報である u と KGC 番号である θ を入力として使用しているのに対し，Lewko の方式では KeyGen において，KGC 番号である i のみを入力として使用しており ユーザの属性情報を使用していない．これは，Lewko の方式では 1 つの KGC では扱うことができる属性が一種類であることを示しており，扱う属性数を増やしたい場合は扱う属性の個数分の Authority Setup を実行して，複数の KGC を管理する必要があることを意味する．そのため，KGC が管理する属性ごとに公開パラメータのサイズが増大する Lewko の方式と，KGC が管理する属性数に依らず公開パラメータが定数サイズで済む Rouselakis らの方式では，公開パラメータの大きさがどの程度変わるのかを比較する．

まず，提案したシステムに登録している全組織の数を N_{KGC} とし，各組織それぞれの属性数が N_{Attr}，であるとする．１つの KGC 当たりの公開パラメータの大きさを D_{Param} とすると，Lewko の方式では公開パラメータのサイズは $N_{\mathrm{KGC}}\times N_{Attr}\times D_{\mathrm{Param}}$ となり，Rouselakis らの方式では $N_{\mathrm{KGC}}\times D_{\mathrm{Param}}$ となる．たとえば，全組織数が $N_{\mathrm{KGC}} = 2^{10}$，属性数がそれぞれ $N_{\mathrm{Attr}} = 2^{10}$，１つの公開パラメータのサイズが $D_{\mathrm{Param}} = 1\mathrm{KB}$ と仮定する．ここで属性数を $N_{\mathrm{Attr}} = 2^{10}$ のように大きな値に設定した理由について説明する．今回比較する 2 つの方式では，安全性証明の条件の都合により，KGC の立ち上げ時の Authority Setup を実行する時点で，KGC が取り扱うことのできる属性の集合を決定する必要があり，Authority Setup 後に扱える属性を増やすことができない．そのため，将来的に利用するかもしれない属性を網羅的に登録する必要があり，各 KGC 内でその時点で扱う属性数が少なかったとしても，登録しておかなければいけない属性数は小さくならないことが想定される．例えば，大学の研究室ごとに KGC を管理する場合，改組などに備え，その大学の全学部・学科が扱う可能性がある属性を用意するような運用方法も考えられる．この場合，KGC に事前に登録しておく属性数は数百となることも考えられる．さらに，第 3.6 節で議論しているユーザの属性変更時の対応（属性鍵の失効）を考慮すると，学生や教職員の退学・異動などに対応するために「2021 年 1 月採用，2022 年 2 月採用」のように，有効期限で制御する．そのため，1 か月単位など比較的短い粒度で属性を付与する必要が生じる．この場合，例えば 10 年間の属性をあらかじめ定義しておくとしても，10 年間で $12 \times 10 = 120$ 個の属性を事前に KGC に登録しておく必要がある．この他にも，プロジェクト毎に属性を分ける場合なども考慮すると，属性数は平均して数百規模にはなると考えており，保守的な評価として $N_{Attr} = 2^{10}$ と見積もっている．

このとき，Lewko の方式では公開パラメータのサイズは $2^{10}\times 2^{10}\times 1\mathrm{KB} \fallingdotseq 1\mathrm{GB}$ となる．一方，Rouselakis らの方式では $2^{10}\times 1\mathrm{KB} \fallingdotseq 1\mathrm{MB}$ となる．公開パラメータはユーザが定常的に持っていなければならないパラメータであり，Lewko の方式では 1GB と非常に大きくなってしまうのに対して，Rouselakis らの方式では 1MB と Lewko の方式と比べて小さいものとなっており，ユーザの負担が Lewko の方式に比べ軽減される．

実際の D_{Param} のサイズについて検討する．今回の実装では，使用している楕円曲線とペアリングから，$\mathbb{G}_T$ の要素のサイズは 3072 ビット（384 バイトで格納），$\mathbb{G}$ の要素のサイズは 1537 ビット（193 バイトで格納）となる．Lewko の方式および Rouselakis らの方式における公開パラメータ D_{Param} をそれぞれ D_{Lewko}, $D_{\mathrm{Rouselakis}}$ と表記する．Lewko の方式の公開パラメータは PK = $\{e(g,g)^{a_i^1}, e(g,g)^{a_i^2}, g^{\vec{b}_{1,i}}, g^{\vec{b}_{2,i}}, g^{\vec{b}_{3,i}}, g^{\vec{b}_{4,i}}\}$ であり，$\mathbb{G}_T$ の要素が 2 個，$\mathbb{G}$ の要素が 4×12 個であるため，$D_{\mathrm{Lewko}} = 2\times 384 + 48\times 193 = 10032$ Bytes となる．Rouselakis らの方式の公開パラメータは PK = $\{e(g,g)^{\alpha_\theta}, g^{y_\theta}\}$ であり，$\mathbb{G}_T$ の要素が 1 個，$\mathbb{G}$ の要素が 1 個であるため，$D_{\mathrm{Rouselakis}} = 1\times 384 + 1\times 193 = 577$ Bytes となる．これは，システム全体の公開パラメータの大きさの差が上記で比較したものよりさらに大きいことを意味している．

これらの比較より，今回の提案システムに Lewko の方式を用いるには公開パラメータが非現実的なサイズとなることが分かった．したがって，今回の提案システムに用いるには，公開パラメータのサイズが現実的である Rouselakis らの方式が適切であることが分かった．しかしながら，公開パラメータのサイズは安全性とトレードオフの関係であり，Lewko の方式で証明されている Adaptive 安全性の方が，Selective 安全性よりも自然なモデルでの安全性となっている．一方，達成している安全性が実用上どれだけの影響を及ぼすかは，著者らが知る限り明らかにされてはいない．Adaptive 安全性の証明がなされている Lewko の方式の方が安全性においては理想的であるが，Selective 安全性しか達成されていない Rouselakis らの方式でも，実際に利用する際には問題がないと考える．以上の結果より，今回の提案したシステムには Rouselakis らの方式を使用するのが現実的である．

3.5.5　MA-ABE の方式ごとの処理時間の比較

第 3.5.4 項では，Rouselakis らの方式が公開パラメータの管理の点で優位であることを確認した．しかし，実際の処理速度が Lewko の方式に比べて著しく遅い場合は，Lewko の方式を使用すること視野に入れる必要が出てくる．そこで，本節では方式ごとの処理速

表 3.2 計測に使用した機器の仕様（許諾を得て文献 [1] より転載）

CPU	Intel®core™i7-920 @ 2.67GHz
Memory	6GB
OS	Debian GNU/Linux 9.5

表 3.3 各アルゴリズムの処理時間の比較 [sec]（許諾を得て文献 [1] より転載）

	Lewko の方式	Rouselakis らの方式
Global Setup	0.0846	0.0840
Authority Setup	1.4901	0.0816
KeyGen	0.6520	0.5379
Encrypt	0.6632	0.4888
Decrypt	1.0172	0.2543

度の違いを比較し，Rouselakis らの方式が Lewko の方式と比べて処理速度が劣っていないことを確認する．

Lewko の方式と Rouselakis らの方式はどちらも対称ペアリングから構成されるため，C 言語用のペアリング暗号ライブラリである PBC Library (version 0.5.14) [*1] の対称ペアリング用の曲線である Type A curve を用いて実装した．ただし，Type A curve として PBC library で提供されている楕円曲線の位数が 160 ビットであり，80 ビット安全性しか有していないため，PairingParametersGenerator API を用いて 256 ビット位数の曲線を生成し，128 ビット安全性を確保している．なお，同様にペアリング計算の出力となる拡大体 $\mathbb{G}_T$ のサイズは 3072 ビットとした．この変更をするためには，PairingParametersGenerator API により Type A Curve で $rBits = 256, qBits = 1536$ を指定して生成をすれば良い．

今回の実装では，暗号化/復号での平文 M として拡大体 $\mathbb{G}_T$ をランダムに取得する実装にしている．

$\mathbb{G}_T$ のサイズは 3072 ビットあるため，128 ビットなどの共通鍵を暗号化するには十分であるため，ハイブリッド型暗号化の公開鍵部分の処理の評価としては十分であると考えられる．

実験での KGC の数は 4 つとし，それぞれの KGC は 1 つの属性を有しているとして暗

[*1] `https://crypto.stanford.edu/pbc/`

号化を行う条件で実験する．各 KGC が有している属性を A, B, C, D という文字としたとき，暗号化の際のアクセス権は A AND ((B AND C) OR D) と固定し，A AND D のユーザの鍵で復号し実験を行った．この条件の一例として，暗号化するデータを閲覧できるユーザを「東海大学」の「総合理工学研究科の学生」または「東海大学」の「教職員」とする場合，上記のアクセス権において，「A」を「東海大学」，「B」を「総合理工学研究科」，「C」を「学生」．「D」を「教職員」と定義することにより，実現することができる．表 3.2 の実験環境において，各アルゴリズムを 100 回実行した平均処理時間を比較したものを表 3.3 に示す．この表より，各 KGC が 1 つの属性を持つという Rouselakis らの方式の利点を生かしていない条件であっても，Lewko の方式に比べ Rouselakis らの方式は処理時間が全てのアルゴリズムにおいて同程度の処理時間，または Lewko の方式よりも早く処理できていることがわかる．以上の結果は各 MA-ABE の方式自体の性質によるものである．前述した通り，Lewko の方式では Adaptive 安全性を実現するために DPVS を用いている．しかしながら，DPVS を用いることにより処理に時間のかかるペアリング演算を含めた全体的な演算回数が，Rouselakis らの方式に比べて非常に多くなっている．そのため，今回の 2 つの方式の処理時間の差は各方式の特性の違いが原因である．

3.5.6　Rouselakis らの方式を用いたファイル共有システムの計測・評価

第 3.5.4 項および，第 3.5.5 項で Lewko の方式と Rouselakis らの方式を比較し，Rouselakis らの方式が Lewko の方式よりも提案システムに使用するのに適していることを確認した．そこで本節では，Rouselakis らの方式を使用してファイル共有システムを構築することを目標とし，実際に Rouselakis らの方式を利用した際の条件式ごとの処理時間を評価する．本節で計測を行っているシステムは，第 3.5.5 項と同様に C 言語用のペアリング暗号ライブラリである PBC Library を用いて実装を行っている．

属性条件による処理速度の計測・評価

アクセス権として設定する論理式の AND と OR の影響について確認するために，属性を AND のみで連結する論理式と，OR のみで連結する論理式の 2 つ条件での暗号化・復号を行い，その処理時間の変化の計測・評価を行った．Global Setup，Authority Setup，KeyGen については，暗号化する際のアクセス権とは関係なく一定の処理速度となることから，表 3.3 の値を採用する．Global Setup は本ファイル共有サービスを開始するときに

初めに 1 回だけ実行する処理，Authority Setup は新たな組織がサービスに参加するとき，KGC を構築するときに 1 回だけ実行する処理である．それぞれの処理は 0.085 秒未満で実行できていることから，十分現実的な処理速度であると考えている．KeyGen はユーザがシステムを利用するときに秘密鍵を取得する際に各 KGC で実行される処理である．ユーザの鍵の取得する頻度によるが，例えば始めに 1 回だけ実行する場合や 1 日に 1 回実行される場合もある．実験結果はユーザが持っている 1 つの属性に対応する鍵の生成に必要な時間である．例えば 10 個の属性を持ったユーザの場合，1 つの KGC から全ての属性に関する鍵を取得するような最悪のケースを考えても，KeyGen の結果の 10 倍の 6 秒程度の時間で鍵が生成されることになる．

属性を AND で連結した場合の結果を表 3.4，OR で連結した場合の結果を表 3.5 に示す．AND 連結で属性数を増やした場合，暗号化（Encrypt）および復号（Decrypt）の処理時間は連結した属性の個数に比例して増加している．OR 連結で属性数を増やした場合は，暗号化は属性の個数に比例するが，復号処理は属性の個数に依らず一定の時間となることが確認できる．暗号化と AND 連結の復号に関しては，属性の個数が増える毎にそれぞれ 0.13 秒未満の処理時間，0.07 秒程度の処理時間が増加しており，OR 連結の復号に関しては 0.07 秒未満で処理できている．Rouselakis らの方式の特性上，暗号化に関しては AND 連結と OR 連結どちらにおいても，アクセス権に規定した属性数分の演算処理を行う．そのため，以上の結果は属性数が増えるごとに線形的に処理時間が増えていっており，これは Rouselakis らの方式の特性通りとなっている．また復号に関しては，AND 連結においては連結されている属性すべてに対する処理を行った結果が必要なため，連結されている属性数分の演算処理が必要となる．そのため，属性数が増えるごとに線形的に処理時間が増えている上記の結果は，Rouselakis らの方式の特性通りである．さらに OR 連結に関しては，連結されている属性のうちどれか 1 つの属性をもっていれば復号可能となる．したがって，連結されている属性数にかかわらず，1 つの属性分のみ演算処理を行うことで復号可能なため，属性数が増えた場合でも一定の処理時間となるのは Rouselakis らの方式の特定通りである．

属性を AND で連結した場合の暗号化・復号処理時間について検討する．属性を AND で連結するのは復号できるグループを制限する場合であり，絞り込む範囲によって連結する個数が決まる．例えば，「ある組織」の「ある役職」の「何年に入社した人」かと言った条件の場合は 3 つの属性が AND で連結させるわけである．通常の使い方では，絞り込む

表 3.4　AND 連結の属性の場合の処理時間 (Rouselakis らの方式) [sec]
（許諾を得て文献 [1] より転載）

属性の個数	1	2	3	4	5
Encrypt	0.1105	0.2457	0.3675	0.4892	0.6071
Decrypt	0.0636	0.1272	0.1907	0.2542	0.3179

表 3.5　OR 連結の属性の場合の処理時間 (Rouselakis らの方式) [sec]
（許諾を得て文献 [1] より転載）

属性の個数	1	2	3	4	5
Encrypt	0.1105	0.2199	0.3283	0.4370	0.5419
Decrypt	0.0636	0.0637	0.0635	0.0635	0.0635

条件数は比較的大きくならないと予想されるため，例えば高々 10 個の属性の AND 連結が上限と想定すると，暗号化は 1.3 秒未満，復号は 0.7 秒未満で処理ができると見積もることができる．

次に，属性を OR で連結した場合の暗号化・復号処理時間について検討する．属性を OR で連結するのは復号できるグループを拡張するときに生じる．共同研究などで 3 つのグループのメンバーがファイルにアクセスすることを認めるとすると，3 つの属性を OR で連結する必要が出てくる．提案システムでは，組織間の比較的大規模な共同研究なども想定しているため，共同でファイルを扱う組織数とその内部でもグループ分けによって，OR で連結される属性の個数は比較的多くなると予想している．例えば，100 グループの属性が OR で連結されることを想定すると，暗号化は 13 秒未満かかることになるが，各グループでの復号処理は 0.07 秒で済むことから閲覧する側は効率的となる．実際は AND 連結して表現されたグループの属性が OR 連結されるため，復号処理は各グループを表現する AND 連結の個数に依存すると思われる．上記の AND 連結の例のように高々 10 個であれば，OR でどれだけ連結したとしても復号処理は 0.7 秒までに抑えられるということになる．

通信処理を含めた処理速度の計測・評価

提案するファイル共有システムの評価をするために，2 種類の実験を行った．1 つ目は KGC の鍵発行に要する処理時間の計測である (図 3.2)．この処理時間はユーザがサービ

スを利用開始する際に生じる時間である．2 つ目は暗号化データの保存/閲覧時間の計測である（図 3.3）．この処理時間はそれぞれ，ユーザが手元でデータを暗号化してストレージへ保存するまでの時間（データの保存時間），ストレージから暗号化データを取得して復号するまでの時間（データの閲覧時間）についての評価をするものである．

提案システムは複数組織間での利用を想定したものとなるため，ユーザとサーバ間での通信によるデータのやりとりが発生する．提案システムでは，第 3.2 節での例「A 大学教員かつ B 大学客員研究員」のように，複数の組織に所属しているようなユーザも想定している．このユーザは自身が主に所属している組織（A 大学）以外にも他組織（B 大学）の KGC にアクセスし，他組織での属性の秘密鍵を発行する必要があるため，ユーザと他組織の KGC 間での通信が発生する．その他にも，ユーザがデータの保存や閲覧などを行う際に暗号化したデータのアップロードや，暗号化されているデータのダウンロードを行う部分において，ユーザとサーバ間での通信が発生する．これらの通信部分は，ユーザ側からのデータは HTTP/HTTPS リクエストを Python と requests ライブラリを用いてサーバに送信するように実装した．サーバ側では，データの保存及び HTTP/HTTPS によるデータの送受信を，Apache 上で動作させた Python と Flask フレームワークを用いた実装を行った．図 3.2, 3.3 はユーザとサーバ（KGC やストレージ）との通信を図示している．評価実験において，ユーザのプログラムは東海大学高輪キャンパス（東京都）に位置しており，サーバはさくらのクラウド (石狩リージョン)[*2]の仮想サーバを利用している．サーバとして上記のクラウドを利用した理由は，応答時間の大きな国内他組織の KGC との通信を模擬できると考えたからである．なお，実験時の東海大学高輪キャンパス・さくらのクラウド（石狩リージョン）間の RTT は試行回数 100 回の平均で 0.022 sec であった．

KGC の鍵発行に要する処理時間

ユーザが KGC に自身の復号用の秘密鍵の生成リクエストを行い，KGC がユーザに鍵を配送するまでにかかる処理時間の評価をする．この処理ではユーザから KGC にリクエストを行う際に，ユーザ自身の ID とパスワードを送信して認証する．KGC はこれを受け取りデータベースに記録されている ID とパスワードのリストと照らし合わせ，認証を行った後にユーザの属性に見合った鍵を生成・配送するといった処理を行っている．なお，KGC は複数の属性に対応する鍵を生成して返すこともできるが，本実験ではユーザ

[*2] https://cloud.sakura.ad.jp/

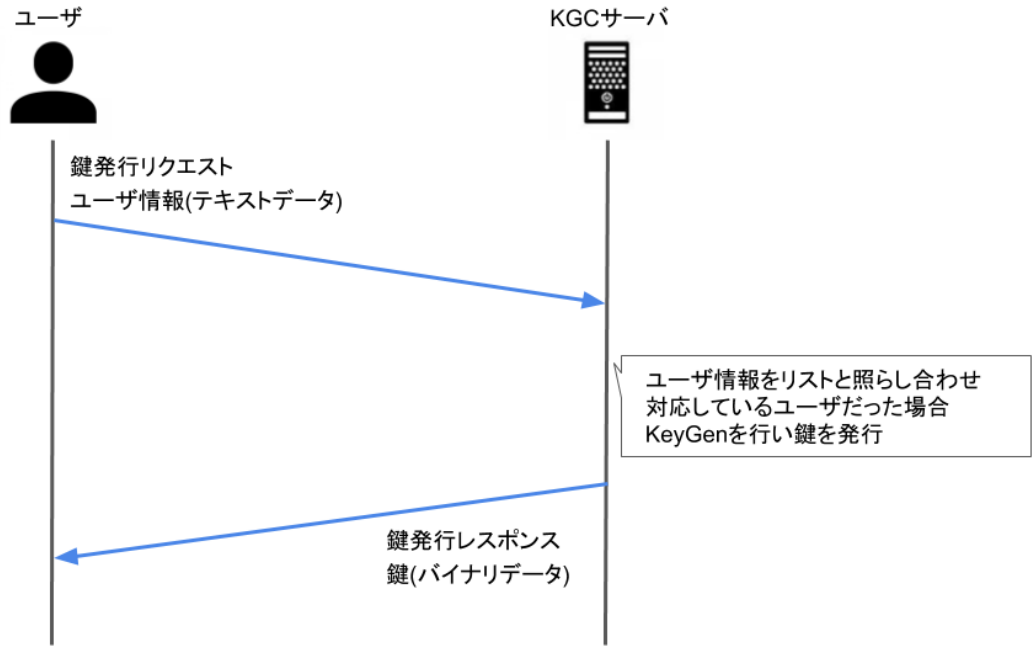

図 3.2　KGC による鍵発行（許諾を得て文献 [1] より転載）

表 3.6　計測に使用したサーバ機器の仕様（許諾を得て文献 [1] より転載）

CPU	Intel®Xeon®CPU E5-2640 @ 2.50GHz
Memory	16GB
OS	CentOS Linux release 7.5.1804 (core)
Python version	Python 3.5.6
Flask version	Flask 1.0.2
Apache version	Apache / 2.4.6

の 1 つの属性に対応する秘密鍵を取得する場合の処理時間を計測している．

表 3.6, 3.7 の実験環境において 100 回実行した平均処理時間を表 3.8 に示す．この表より，鍵発行に関する処理は全体で 0.31 秒程度で行えることが分かった．

データの保存および閲覧にかかる処理時間

ユーザがデータを暗号後にサーバ上にアップロードするまでの処理時間と，ユーザが暗号化されたデータをダウンロードし，復号するまでの処理時間の計測をそれぞれ行った．

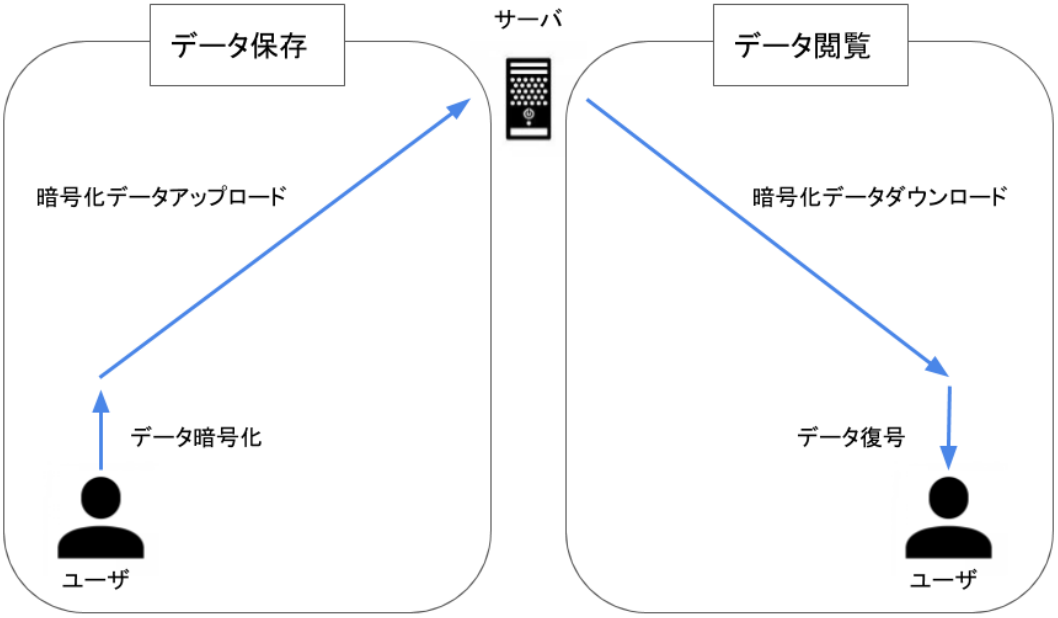

図 3.3　データの暗号化および復号（許諾を得て文献 [1] より転載）

表 3.7　計測に使用したユーザ機器の仕様（許諾を得て文献 [1] より転載）

CPU	Intel®core™i7-6950X @ 3.00GHz
Memory	64GB
OS	Debian GNU/Linux 8
Python version	Python 3.4.2
Requests version	Requests 2.4.3

表 3.8　鍵発行にかかる通信部分を含めた処理時間 [sec]（許諾を得て文献 [1] より転載）

	鍵生成	通信時間	合計
処理時間	0.1907	0.1176	0.3083

今回の実装では Authority Setup や KeyGen，Encrypt の際に生成されるパラメータはそれぞれ，バイナリファイルとして出力されるようになっており，それらのパラメータを使う処理を行う際には，それらのファイルからパラメータを入力して使用する仕様となって

表 3.9　暗号化に関する部分の処理時間 [sec]（許諾を得て文献 [1] より転載）

	暗号化	送信	合計
処理時間	0.4888	0.1176	0.6064

表 3.10　復号に関する部分の処理時間 [sec]（許諾を得て文献 [1] より転載）

	受信	復号	合計
処理時間	0.1299	0.2543	0.3842

いる．出力されるファイルのフォーマットは，1 行目には格納される要素のサイズを水平 tab 区切りにてアスキーコードで書き込み，2 行目以降には各要素をバイナリで書き込む．例えば Authority Setup では，KGCθ は公開パラメータ PK = $\{e(g,g)^{\alpha_\theta}, g^{y_\theta}\}$ といった 2 つのパラメータを公開するが，この場合 1 つ目の要素 $\{e(g,g)^{\alpha_\theta}\}$ は $\mathbb{G}_T$ 上，2 つ目の要素 $\{g^{y_\theta}\}$ は $\mathbb{G}_1$ 上の点となるためのパラメータの大きさはそれぞれ 384，183 ビットとなるため，1 行目には「384（水平 tab）183」とアスキーコードで出力し，改行をした後にバイナリでそれぞれのデータを格納した．他のパラメータも同様に，出力されるパラメータ数に応じてバイナリファイルに出力されるように実装を行った．

実験では KGC の総数は 4 つとし，それぞれの KGC は 1 つの属性を有しているという条件で計測する．各 KGC が有している属性を A, B, C, D という文字としたとき，暗号化の際のアクセス権は A AND ((B AND C) OR D) と固定し，A AND D のユーザの鍵で復号するといった条件で実験を行った．

まず初めに，データの保存に要する処理時間を計測する．データの保存の際に行っているクライアントでのデータの暗号化，およびクラウドサーバへのアップロード処理の処理時間を表 3.9 に示す．この処理時間は，表 3.6, 3.7 の実験環境において 100 回実行した平均値としている．実験結果より，データ保存時の全体の処理時間は 0.61 秒程度で行えることが分かった．そのうち通信にかかる時間は 0.12 秒程度であった．

次に，データの閲覧に要する処理時間を計測する．データの閲覧の際に行っているクラウドサーバからの暗号化データのダウンロード，およびクライアントでのデータの復号処理の処理時間を表 3.10 に示す．この処理時間は，表 3.6, 3.7 の実験環境において 100 回実行した平均値としている．実験結果より，データ閲覧時の全体の処理時間は 0.39 秒程

度で行えることが分かった．そのうち通信にかかる時間は 0.13 秒程度であった．

3.6 ユーザの属性変更に関する考察

MA-ABE を利用したシステムを実際に運用する場合，ユーザの昇進，部署移動，卒業，退学などに基づいた，ユーザの属性変更による鍵失効を考慮する必要がある．ユーザはその時点での自身の属性に基づく秘密鍵を KGC から認証を経て発行後に受け取るが，その後にユーザの属性が変更になった場合について MA-ABE 方式自体は想定していない．そのため本節では，これらの場合に望まれる属性変更への対応方法について考察を行う．

3.6.1 鍵失効を実現する方法

今回提案したファイル共有システムを実際に使用する場合，ユーザの属性変更が生じた際に，ユーザの元の属性の秘密鍵が失効され，属性が変更したユーザは変更前の属性の鍵を使ってコンテンツの復号が行えなくなるといったことが望まれる．そこで本節では，鍵失効を実現するにあたっていくつかの方法を考案し，実際のシステムでの利用に適した方法について考察を行った．

鍵失効機能を有する ABE を利用する方法

単一の KGC が存在可能な ABE に対する失効方式は，既にいくつか提案されている．まず，正規ユーザへの更新鍵配布を用いた Indirect な失効方式 [26] [27] [28] が挙げられる．これは，ある期間 T ごとに更新鍵の発行を要する代わりに，暗号文作成者に失効情報を意識させることなく，失効を実現できる．

次に更新鍵を用いない Direct な失効方式 [29] が存在する．これは暗号文作成者が失効情報を知り得る場合の失効方式で，暗号文作成時に暗号文に対して直接失効者を規定する方法である．また，Indirect と Direct の両方の失効方式を備えた方式 [30] も存在する．さらに，公開パラメータが定数サイズで抑えられる鍵ポリシー属性ベース暗号 (Key-Policy Attribute-Based Encryption: KP-ABE) に対して，失効機能を実現した方式 [31] が存在する．しかし，単一の KGC が存在可能な ABE の利用は複数組織で利用可能なファイル共有システムに適していない．

一方で，MA-ABE に対する失効方式も提案されている．MA-ABE に対して Direct な失効機能を実現した方式 [32] や，失効情報が含まれるパッチをあるユーザ属性の失効ご

とに発行し，MA-ABE 暗号文に適用する失効方式 [33] が存在する．また，中央機関が存在する MA-ABE に対して効率のよい失効機能を実現した方式 [34] や，CCA 安全を達成しつつ，公開パラメータが定数サイズに抑えられる MA-ABE に対して失効機能を実現した方式 [35] も提案されている．

複数組織間で利用可能なファイル共有システムにおける属性管理を考える上で，文献 [35] の方式を用いる方法が良いように思える．しかし，文献 [35] の方式は著者らの知る限り実装が存在しない．また，文献 [35] の方式は Indirect な失効方式であるため，更新鍵の配布が必要となる．更新鍵を配布する場合，実際にシステムを運用する上で，どの程度のコストを要するか検討する必要があるが，文献 [35] ではそのような検討は成されていない．

文献 [1] 発表時には，提案システムに適する鍵失効機能を有する MA-ABE が存在していなかったが，SecITC2022 にて提案システムに適していると考えられる Large Universe かつ鍵失効機能を備えた MA-ABE の提案がされている [36]．これは方式自体が鍵失効機能を有しており，ユーザの鍵に有効期限などを埋め込むなどの工夫をすることなく，必要なタイミングで MA-ABE の方式自体の機能によって，鍵失効を行うことができる方式となっている．今回の提案システムの実装・評価を行う際はこの MA-ABE の方式を用いて行うことが有益である可能性があるが，この方式を用いた実装・評価を含めた検討は今後の課題とする．

代理人再暗号化方式を用いる方法

代理人再暗号化方式と MA-ABE を組み合わせる方法が挙げられる．ここで，代理人再暗号化方式とは，代理人再暗号化サーバが管理する再暗号化鍵によって，ある暗号文を復号せずに，異なる鍵による暗号文に再暗号化することを実現した方式である．失効機能を実現する方法として，暗号文作成者はまず平文を共通鍵暗号方式で暗号化する．次に暗号文作成者は，共通鍵暗号方式における暗号化鍵（=復号鍵）を代理人再暗号化方式で暗号化する．このときアクセス権 A で暗号化した暗号文を CT_A とする．そして暗号文作成者は，代理人暗号化方式における秘密鍵 sk_A を MA-ABE で暗号化する．この暗号文を $CT_{\mathbb{A}}$ とする．

ここで，属性の失効が生じたとき，暗号文作成者は失効情報を各 KGC から得られるものとする．次に暗号文作成者は，CT_A を $CT_{A'}$ に再暗号化するための再暗号化鍵 $rk_{A,A'}$

を生成し，代理人再暗号化サーバへ送付する．代理人暗号化サーバは $rk_{A,A'}$ を用いて，CT_A を $CT_{A'}$ に再暗号化する．さらに，暗号文作成者は $sk_{A'}$ について，失効情報に基づき MA-ABE によって暗号文 $CT_{\mathbb{A}'}$ を生成する．このとき，$\mathbb{A}'$ は失効した属性を含まない新たな条件式で，失効されたユーザは $CT_{\mathbb{A}'}$ を復号できないことに注意する．

組み合わせる代理人再暗号化方式としては，共通鍵暗号ベースの方式 [37] が挙げられる．この方式を用いると，CT_A や $CT_{\mathbb{A}}$ を効率よく生成できる．ここで，文献 [37] を用いたときに，悪意ある復号者が $sk_A, sk_{A'}$ を失効されたユーザに対して，不正配布することが懸念される．$sk_A, sk_{A'}$ を不正配布することと，平文そのものを不正に配布することとでコストに差が無い限り，このような攻撃は問題とならない．一方で，平文の不正配布について，鍵失効に限らず MA-ABE を用いる以上同様の問題が生じてしまう．このため，鍵の失効や属性管理に関しても，悪意のある復号者による不正配布攻撃について，本論文では論じないこととする．

なお，前述の不正配布攻撃に関して論じないとしたとしても，代理人再暗号化方式を用いる方法では，代理人サーバの運用コストが新たに生じてしまう．このため，代理人再暗号化方式を用いる方法は好ましくない．

他のエンティティを用いる方法

サーバにユーザの属性鍵を保存し，アクセス制御を行う方法が挙げられる．先行研究として，専用エンティティが秘密鍵の世代を管理することで失効を実現する方式 [38] が提案されている．また，IC カードにユーザの属性鍵を格納し，IC カード単位で失効を行うという方法も挙げられる．しかしこれらの方法は，暗号化方式単体でのアクセス制御を実現するといった，ABE の利点を損なうことになる．

鍵の属性に有効期限を埋め込む方法

ユーザに鍵を発行する際に，そのユーザの属性が使用される期間があらかじめわかっている場合，鍵の属性自体に有効期限を設定する方法が考えられる．たとえばユーザの属性が「学生」だった場合を考えると，ユーザが大学の学部生だった場合，その鍵を使用する期間は 4 年間が想定される．そこで鍵に埋め込む属性を「学生（2018 年度〜2021 年度）」のように有効期限を設定することにより，このユーザが卒業した後の暗号文に対して鍵の失効が可能となると思われる．また，企業で 1 年ごとに部署移動や昇進などの属

性変更が行われるような組織で使用される場合は，鍵に埋め込む有効期間の部分を「人事部課長（2018 年度）」のように 1 年ごとなどにする．こうすることによって比較的短い期間でユーザの属性変更が行われる組織の場合でも，この方式で対応することが可能になると思われる．この方法を用いる場合には，一定期間ごとにユーザの鍵の再発行が必要となることが想定される．しかしながら，本章にて提案しているファイル共有システムでは KGC を研究室単位などの比較的小さな組織毎に管理をすることを想定している．そのため KGC 当たりの管理するユーザ数が比較的少なく抑えられるため，鍵に有効期間を設定した場合でも，鍵の再発行にかかる負荷は許容できる程度になると考えられる．そのため，実システムを運用していく上では，属性管理に関して鍵の属性に有効期限を埋め込む方法が有用と思われる．

3.7　複数組織間の運用に関する考察

運用面では，先述したユーザの属性変更と鍵の運用だけではなく，組織間で暗号システムを運用する際の課題も存在する．本節では複数組織間の運用においてどういった課題が生じうるかを整理し，その対応の方向性を議論する．

3.7.1　提案ファイル共有システム運用開始前の組織間合意

本研究のシステムでは複数の組織がそれぞれ KGC を運用する．この際，組織間で KGC の運用レベルが異なる場合に暗号システムとしての問題が起こりうる．そのため，運用の開始前にどういった運用レベルでそれぞれの KGC が運用されるかをお互いに確認し合意を取る必要がある．たとえば，KGC の運用において電源やネットワークについて耐障害性がどれほど整備されているか，KGC の秘密情報（秘密鍵）をいかに保管するか，秘密情報のバックアップを取るかどうか，各ユーザに向けて発行された鍵の情報をどう保管するか，などがある．さらには，KGC を運用する組織の体制についての相互の確認や，障害により問題が発生した場合の補償についても考えうる．これらを踏まえた合意を取ることで，提案システムは適切に運用されることになる．合意形成についてはさらに深く検討が必要であるが，本研究ではシステム構成の提案に主を置くため，運用にあたっての合意形成プロセスについては必要性の検討に留めた．

3.7.2 提案ファイル共有システムのクラウドコンピューティング環境での運用

本提案システムでは，KGC の運用を柔軟に行えることが可能であるため大学の研究室等の小規模組織で KGC を運用管理することを想定している．しかし原理的にはクラウドコンピューティング環境に KGC を配置しても暗号化や復号，鍵発行といった作業は問題なく行える．

一方で，クラウドコンピューティング環境に置くことでマルチテナントという性質などが本提案システムに何らかの影響を与える可能性はある．たとえば各 KGC の秘密情報（秘密鍵）の管理が組織内端末を想定しており KGC 内部では保護されずに保存されている場合には，KGC の秘密情報をクラウドコンピューティング管理者が取得し扱うことができてしまう．通常の暗号システムであれば，秘密鍵を利用した作業を実施するにあたり秘密鍵に施されたパスワードやハードウェアトークンによる認証により保護機構を外しアクティベートすることで秘密鍵を保護しているが，そういった機構の利用は本提案のシステム構成としては必須とはなっていない．その他にも，クラウドコンピューティングであることで組織内の物理的なコンピュータとは異なる運用をしなければならず，それが暗号システムの安全性に影響を与える事象の存在は十分に考えられる．クラウドコンピューティング環境を利用する場合には，それらを整理し対応することが必要になる．これらを検討し対応することで，少なくともシステムの全利用組織が単一の KGC を信頼する従来の CP-ABE システムよりは柔軟かつ安全に利用可能であると考えられる．

3.8 結言

本章では，複数組織対応属性ベース暗号（MA-ABE）を用いたファイル共有システムについて，具体的なユースケースを想定した上で実現可能性について考察した．まず，第 2.5.3 項にて今回の提案システムには以下の要件を満たすことが必要であると分かった．以下の要件 1〜4 は，MA-ABE を用いる方法以外のファイル教諭システムを設計・運用する場合にも適用可能な要件となっており，暗号システムを実際のシステムとして安全かつ実運用する際に広く利用できる汎用的な要件が明確化された．

- **要件 1：複数組織間で利用可能なファイル共有システムであること**

- **要件 2：共有するファイルを，共有元のユーザが指定する相手のみが編集・閲覧でき，サービス提供者も含めて他者にはファイルの内容がファイル名・ディレクトリ名も含めて漏洩しないこと**
- **要件 3：ユーザの鍵管理が容易であること**
- **要件 4：ファイルを共有するユーザの指定を動的に変更できること**

以上の要件に沿って，今回想定している研究データ向けファイル共有システムに適している既存の MA-ABE 方式について調査・比較を行った．その結果，高度な安全性を重視するのであれば Lewko の方式 [10]，データサイズや処理性能など実用性を重視するのであれば，Rouselakis らの方式 [9] が想定する提案システムに適していることが分かった．次に，Lewko の方式と Rouselakis らの方式の公開パラメータサイズ，および処理時間について具体的な数値による比較・検討を行った．その結果，安全性の違いはあるものの公開パラメータのサイズに関して，提案システムに用いるには Rouselakis らの方式の方が適していることが確認できた．この 2 つの方式においては，実装時の各アルゴリズムの処理時間においても Rouselakis らの方式の方が速いことが分かった．以上の結果から，今回の提案システムには Rouselakis らの方式が適していることが示された．

Rouselakis らの方式が提案システムに適していることを確認したため，次に，この方式を用いた提案システムにおける詳細評価として，複数の暗号化条件による処理時間および通信処理を含めた評価を行った．一番処理回数が多いと考えられる暗号化・復号に関して，属性数毎に処理時間の変化を計測したところ，OR 連結で属性数を増やした場合は暗号化は属性の個数に比例するが，復号処理は属性の個数に依らず一定の時間となることが示された．OR 連結の復号に関しては 0.07 秒未満で処理でき，AND 連結の復号に関しては，属性の個数が増える毎に 0.07 秒程度の処理時間が増加することが分かった．また，どちらの連結も暗号化は 0.13 秒未満の処理時間が属性の個数ごとに増加していくことが確認できた．さらに通信時間を含めた評価を行った結果，鍵発行センターからユーザが秘密鍵を取得するまでの通信時間を含めた処理時間は，単一の属性の鍵では 0.31 秒程度，暗号化データの保存/閲覧に要する時間は保存時には 0.6 秒程度，閲覧時は 0.4 秒程度であることも判明した．以上の結果より，Rouselakis らの方式の MA-ABE を提案システムに用いることにより，提案システムが満たすべき要件 1〜3 と要件 4 における鍵失効以外の部分に関して，満たすことが可能であることを示した．

最後に，実際に提案ファイル共有システムを運用する場合に，部署異動などでユーザの権限が変化した場合に対応する方法，具体的にはユーザの属性の変更時の鍵失効方法についての考察を行った．鍵失効機能付き MA-ABE を使用した方法，代理人再暗号化と MA-ABE を組み合わせる方法，他のエンティティを用いる方法や鍵の属性に有効期限を埋め込む方法について考察した．その結果，ユーザの属性情報に加えてユーザの属性の有効期限も同時に設定することにより，鍵の失効を実現する方法が現在知られている中では最も現実的であるとの結論に至った．以上の方法により，提案システムにおける要件 4 の鍵失効に関する要件を満たすことが可能になることを示した．また，今後の課題として鍵失効機能を有する MA-ABE を提案システムに用いる場合の実装・評価を行うことが必要であることが示された．

本章により，実利用環境を十分に考慮した MA-ABE を用いた研究データ向けファイル共有システムが，満たすべき要件が明確になり，これらの要件を満たしたシステムの構成方法を提案した．実用的な視点でのシステム評価や運用課題の抽出と考察を行い，第 2.5.3 項にて抽出された，提案システムを運用する際に必要となる要件を満たす方法を示すことにより，安全かつ実用的なファイル共有システムの実用性が明確化された．

第 4 章

研究データ向けファイル共有システムの実装・運用に関する設計

4.1　緒言

第 3 章では MA-ABE を用いて複数組織間で利用可能とする，研究データ向けファイル共有システムを実現する方法を示した．そのため，提案システムに用いるのに最適な MA-ABE の方式の選定や，属性管理，鍵失効時の処理方法についての考察など，実用的なユースケースを十分に考慮した様々な検討・提案を行った．提案システムを実運用する際に，システムのセットアップ後に行われる処理のユースケースは以下の 3 つになる．

- **ユーザの鍵発行**
  ユーザが自身の属性情報が含まれた鍵を KGC にて発行し受け取る．ユーザの属性が新しく作成された属性の場合，新規に属性を登録する処理が生じることが考えられる．昇進などにより新たな属性に変更になるユーザの場合，その前に利用していた属性の鍵を失効する必要も考えられる．

- **データの閲覧**
  ユーザはシステム上に保存されているデータの中から，自身の属性で復号可能であるものを復号し閲覧する．この時，データの内容は MA-ABE により暗号化されているため安全性が保たれているが，ファイル・ディレクトリ名も同様に復号可能でないユーザからは閲覧できない．

- **データの編集・保存**
  ユーザはデータを編集し保存する際，そのデータを閲覧できるユーザの属性を決めて MA-ABE を用いて暗号化し保存を行う．この時，データは編集・保存できる権限をもっているユーザでなければこれらの処理を行うことができない．

上記のユースケースを実システム上で処理を行うためには，第 3 章で示した要件を満たすため，Rouselakis らの方式の MA-ABE を提案システム用いること等に起因する，新たに満たすべき要件が存在する．

本章ではまず，先行研究にて提案されている CP-ABE を用いた従来のファイル共有システム [2] の手法を参考に，第 3 章で提案した複数組織で利用可能な研究データ向けファイル共有システムを，実際に運用する際に必要となる要件を新たに定義し，それらの要件を満たす方法を示す．次に，本章で新たに定義した要件と第 2.5.3 項にて示した要件を，それぞれ満たした提案システムの実際の処理手順や，様々なフォーマットに関して詳細設計を行うまた，今回設計した処理手順を実際に想定されるユースケースを例に動作を確認する．これにより，提案システムを実運用する際に設計した処理において，提案システムが満たすべき要件を全て満たしていることを示す．以下に第 2.5.3 項にて明らかになった要件と，第 3 章にて示したその解決方法を再度示す．

- **要件 1：複数組織間で利用可能なファイル共有システムであること**
  複数組織間で利用可能なファイル共有システムを実現するため，既存の CP-ABE を用いたファイル共有システムを参考に，MA-ABE を用いることによって実現方法を示した．
- **要件 2：共有するファイルを，共有元のユーザが指定する相手のみが編集・閲覧でき，サービス提供者も含めて他者にはファイルの内容がファイル名・ディレクトリ名も含めて漏洩しないこと**
  既存の CP-ABE を用いたファイル共有システムを参考に，MA-ABE を用いることにより指定したユーザ以外にファイル名・ディレクトリ名を含めたファイルの内容が漏洩しない方法を示した．
- **要件 3：ユーザの鍵管理が容易であること**
  MA-ABE を用いることにより，この要件を満たすシステムとなることを示した．

- **要件 4：ファイルを共有するユーザの指定を動的に変更できること**
  ユーザの属性変化などに対応するために必要となる鍵失効方法に関して示した．

最後に，提案システムと同様の機能を持つファイル共有システムを，Rouselakis らの方式以外の MA-ABE を用いて構成する場合の実現可能性について考察する．

4.2 MA-ABE を用いたファイル共有システムの詳細設計

本節では，第 3 章にて提案を行った複数組織間で利用可能なファイル共有システムを，実利用可能なシステムとするため，第 4.1 節にて必要であることが示された処理部分の考察・設計を行う．提案システムでは MA-ABE を用いて CP-ABE を用いた従来システム [2] を参考に，ファイル名・ディレクトリ名を含むディレクトリ構造全体を暗号化し，編集権限の制御も行う．MA-ABE を用いて実現している第 3 章で提案したシステムを運用する場合，第 2.5.3 項で明らかになった要件を満たすためには，さらに満たすべき要件が新出する．これらの問題点やシステムの運用課題に関しての詳細を，要件 1〜4 に照らし合わせた上でそれぞれ以下に示す．

まず初めに，「GID は全組織の全ユーザで固有であること」についてである．要件 1 を満たすため，提案システムでは MA-ABE を用いて複数組織で使用可能なファイル共有システムの実現を行っている．そのため，MA-ABE において KGC がユーザの鍵を生成する際に使用する GID を，提案システムを利用する全組織のユーザ全員がそれぞれ固有になるように管理する必要がある．そこで，GID の管理を含めた KGC の管理に学術認証フェデレーション（学認）[*1] を用いる．学認を用いた KGC の管理方法の詳細は第 4.2.1 項にて説明する．

次に，「ファイル名・ディレクトリ名を含めた暗号化ができること」である，参考にしている CP-ABE を用いた従来方式は単一組織で利用が想定されるファイル共有システムである．従来方式ではリストファイルを用いて要件 2 を満たしているが，提案システムは複数組織で利用可能なファイル共有システムであるため，リストファイルのフォーマットに関しては CP-ABE を用いた従来システムの方式を参考に，再構成を行う必要がある．そこで，リストファイルのフォーマットに関して第 4.2.2 項にて考察・提案する．

その次に，「ファイルの編集・保存を権限のないユーザが実行できないこと」である，リ

[*1] https://www.gakunin.jp/

ストファイルの管理の際に必要となるアップロードマネージャによって従来システムでは要件 2 が実現されている．このアップロードマネージャの設置方法に関しても，複数組織で利用可能な提案システムにおいて適する方法を検討する必要がある．そこで，提案システムにおけるアップロードマネージャの設置方法における考察・提案を第 4.2.3 項にて行う．

最後に，「新規属性の鍵発行および鍵失効方法ができること」である，MA-ABE を用いる際の属性管理は，各組織間においてある程度の取り決めを行い，属性登録や鍵の失効などを行う必要があると考えられる．第 3 章では鍵の失効方法について検討した．しかし，提案システムに用いる Rouselakis らの方式の MA-ABE では，システムのセットアップ後に新たに属性を登録することができない．これは満たしている安全性証明に起因するものである．実際のシステムとして運用する場合にはこのようなユースケースが想定されるため，属性管理の方法について，第 4.2.4 項にて考察・提案を示す．

以上の要件 1〜4 に関する新たな要件をそれぞれ要件 A〜D とする．要件 A〜D について以下に整理する．

- **要件 A：GID は全組織の全ユーザで固有であること**
  提案システムでは KGC は組織毎に管理を行うが，MA-ABE において鍵発行の際に必要となる，全組織の全ユーザが固有となる GID は，全組織で統一して管理を行うことにより，全ユーザ固有としなければならない．
- **要件 B：ファイル名・ディレクトリ名を含めた暗号化ができること**
  データ自体の暗号化のみでなく，権限のないユーザにファイル名・ディレクトリ名が見えることによってファイルの内容が漏洩することを防ぐため，ファイル名・ディレクトリ名に関しても暗号化を行い，秘匿される必要がある．
- **要件 C：ファイルの編集・保存を権限のないユーザが実行できないこと**
  ファイルの編集・アップロード時を権限のないユーザによって，データが編集・削除することができないようになっている必要がある．
- **要件 D：新規属性の鍵発行ができること**
  ユーザの鍵発行時に大学の改組などにより，属性の新規作成が必要になった場合，Rouselakis らの方式では後から属性を追加できないため，解決方法が必要となる．

以上の要件は，ファイル共有システム全般に対して満たす必要がある要件となる．本章で

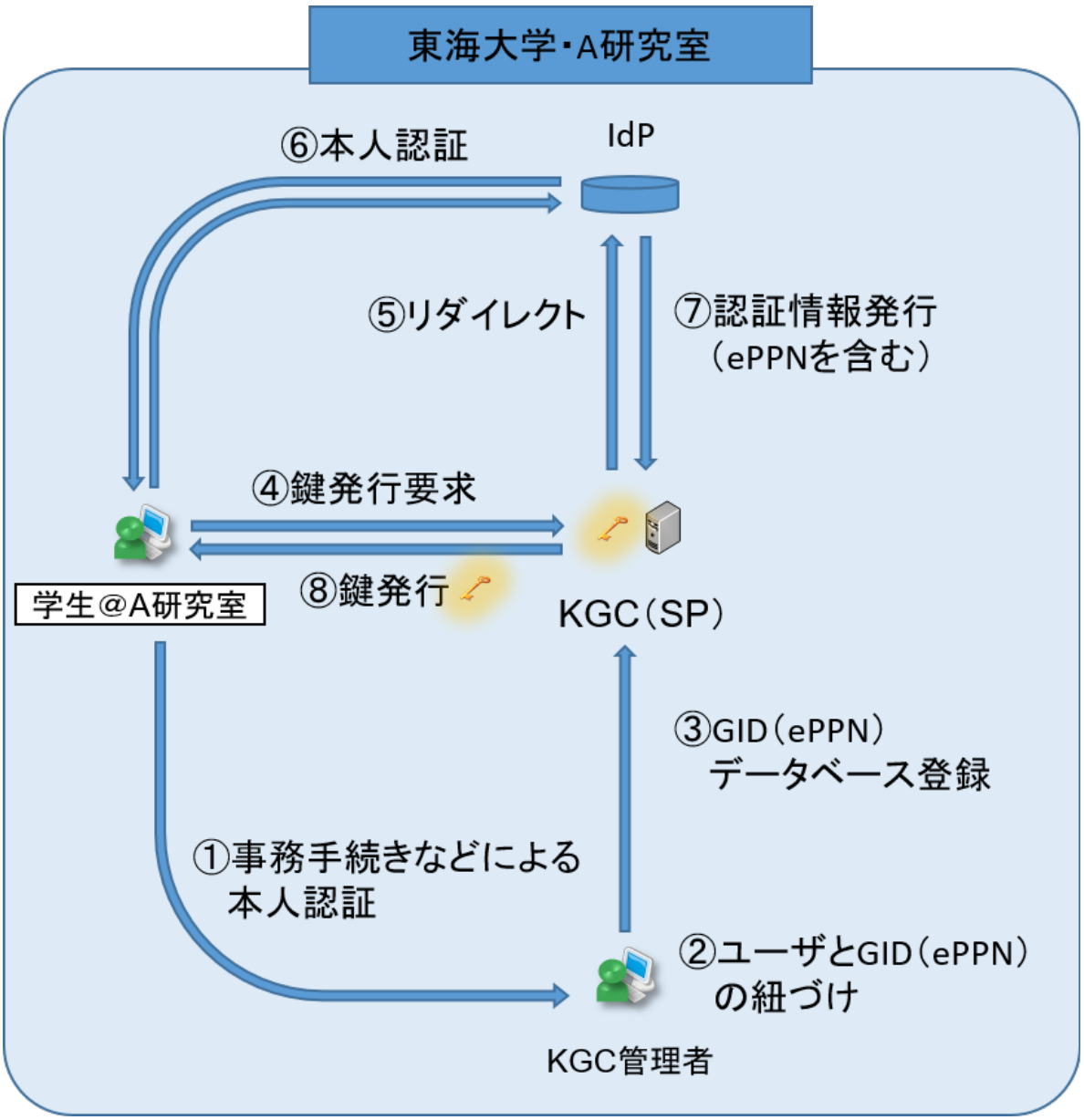

図 4.1　学認を用いた鍵発行の概要（許諾を得て文献 [3] より転載）

は，これらの要件に関する考察を提案システムについてのみ行うが，今後 MA-ABE を用いる方法以外でファイル共有システムを設計・運用する場合にも，これらの要件を満たすシステムであるかどうかを考慮する必要がある．

4.2.1　KGC の運用方法

提案システムでは，暗号用の鍵の管理 (KGC の管理) とユーザの ID 管理（GID の管理）を分けており，後者は各大学等で管理されている ID 管理基盤を利用することを想定している．しかしながら，各組織が独立に ID を管理する場合，GID を自由に付与することができ，第 3.2 節で述べた結託攻撃への防御が困難となる．したがって，異なる組織の ID 管理基盤が管理していても全ユーザで ID が異なるような GID の固有性を実現できる仕組みが必要となる．そこで，本システムでは上記の条件に合致したものとして，学術認証フェデレーション（学認）で用いている ePPN（eduPersonPrincipalName）を利用することを想定する（図 4.1）．ePPN は「ユーザ ID@所属大学のドメイン名」のような形で構成されており，学認においてユーザを識別することのできる加盟組織内で固有の名前になっている．学認の仕組みを用いて提案システムを実現する場合，まず初めに，自組織の認証プロバイダ（IdP）で認証を行い認証情報（ePPN を含む）を発行する．次に，発行された認証情報を利用して，ユーザは鍵発行を依頼する組織の KGC（学認のサービスプロバイダ（SP）として構築する）で属性に対応する秘密鍵の発行を行う．このとき，認証情報に含まれる ePPN を利用することにより，ユーザ本人の GID に対応する属性の秘密鍵を発行することが可能となる．

KGC から発行される秘密鍵に対応する属性の管理については，第 4.2.4 項の方法にて行う．KGC を運営している組織では，所属しているユーザの各 GID に紐づける属性を事前にデータベース等に登録しておき，鍵生成の要求があったときに対応する秘密鍵を払い出す．この登録については，当該組織で属性を持つ場合に何らかの事務手続きが生じると思われるため，その際に本人確認および対応付ける属性を精査した上で登録する．なお，ここで扱う属性の種類については，第 4.2.4 項の方法にて管理を行うため，学認などが提供しているような統一した属性を用いるなど，組織間で事前に取り決めを行う必要がある．その際，組織間で異なる属性を 1 つの属性で管理するなど，属性に範囲を持たせる場合が考えられる．この場合は属性がどの範囲までを包括するのか事前に詳細に取り決めを行い，範囲の誤解が生じないように注意する必要がある．以上の方法より，提案システムにおける要件 A の「GID は全組織の全ユーザで固有であること」を満たすことが確認できた．

この方法では，上記のような GID と属性の紐づけなどの管理を研究室単位等で行わな

ければならないため，研究室の責任者等の KGC 管理者に一定の負担が生じることとなる．しかし，KGC の管理対象が研究室単位等まで小さくなったことから，比較的管理の負荷は小さくなる．

4.2.2　リストファイルを用いたファイル名・ディレクトリ名の管理方法

一般的なファイル暗号化システムではデータの暗号化のみを行うが，提案システムでは CP-ABE を用いた従来システム [2] の手法であるリストファイルという概念を用いて，データの暗号化だけでなくファイル名・ディレクトリ名の暗号化および，編集権限の制御を行う（図 2.4）．これにより，ファイル名などからデータの内容の情報が漏洩することを防止する．

今回参考にしている CP-ABE を用いた従来システムの方式ではデータへのアクセス権のうち，データの又はファイル名・ディレクトリ名を復号できる権限を read 権とし，これらの作成や編集をすることのできる権限を write 権と定義している．そこで提案システムにおいても同様に定義を行う．提案システムでは read 権による制御は MA-ABE を用いて暗号化することで実現する．ファイルのデータ自体は MA-ABE で暗号化し，ファイル名はユニークな疑似乱数列で表される保存用ファイル名に置き換えて保存を行う．このとき，保存用ファイル名と元のファイル名の対応関係に関して MA-ABE で暗号化を行う．これにより read 権がないユーザからファイル名を秘匿することができる．write 権に関しては CP-ABE を用いた従来方式で提案されているリストファイルとアップロードマネージャーマネージャによって制御を行う（図 2.4）．アップロードマネージャはリストファイル内の write 権に対応する属性をユーザが保有しているかを認証する．認証に成功した場合にのみ，アップロードマネージャはストレージへのファイルのアップロードおよびリストファイルの更新を行う．ユーザはこのアップロードマネージャを介してのみファイルのアップロードを行えるようにすることにより，権限のないユーザによる不正なデータの書き換えやファイル削除を防いでいる．ユーザはデータのダウンロードはストレージから直接行い，アップロードはアップロードマネージャを介して行う．この時のアップロードやダウンロードの処理に関しては，ファイルビューアを用いて行う．

CP-ABE を用いた従来システムでは単一組織での利用が想定されているが，提案システムは複数組織で利用可能である．そのため，リストファイルのフォーマットの大部分は CP-ABE を用いた従来システムを参考とするが，フォーマットを一部再考する必要が

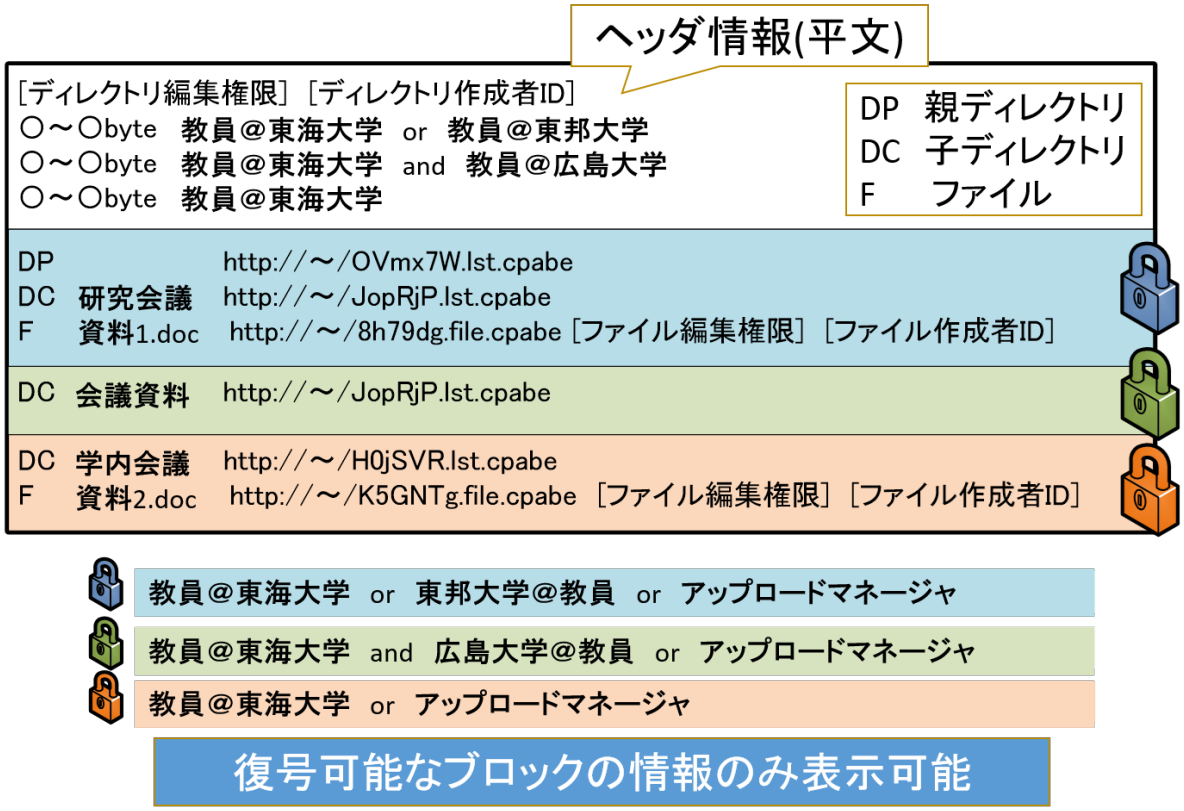

図 4.2　リストファイルの概要（許諾を得て文献 [3] より転載）

ある．提案システムにおけるリストファイルでは，まずヘッダ情報の 1 行目は CP-ABE を用いた従来システムと同様にカレントディレクトリの write 権を表す文字列およびディレクトリ作者の ID を記す（図 4.2）．2 行目以降も同様にリストファイル内の read 権毎のブロックの開始位置をバイト数で表記を行い，その後ろにそのブロックの read 権を表す文字列を格納する．この時，提案システムは複数組織で使用可能であるため read 権に自組織以外のユーザの属性が入る場合がある．そういった場合には，自組織以外の属性が含まれている read 権のブロックを優先的にリストファイル上部から追加していく．そしてその下に自組織の属性のみのブロックを格納していく．以上のように処理を行うことにより，提案システムにてリストファイルを効率的に作成・復号することができ，ファイル名・ディレクトリ名の暗号化および，編集権限の制御を行うことが可能となる．以上の方法より，提案システムにおける要件 B の「ファイル名・ディレクトリ名を含めた暗号化ができること」を満たす方法を示した．

4.2.3 アップロードマネージャ運用方法と設置方法

提案システムでは第 4.2.2 項で設計した通り，リストファイルを用いることによりデータの暗号化だけでなく，ファイル名・ディレクトリ名の暗号化および，編集権限の制御を行う．リストファイルを管理するには共有の属性などを作成し管理することが可能であるが，この場合，共有するユーザによって勝手にファイルの編集をされてしまう可能性がある．そこで CP-ABE を用いた従来システムではアップロードマネージャを用いて集中管理する方法を用いている．アップロードマネージャでは，ファイルの編集権限のあるユーザによる要求が行われた場合にのみストレージへの書き込みを行なう．この際，編集権限の管理にはリストファイルを用いる．ファイルのダウンロードはストレージから直接行うことでアップロードマネージャの負荷の軽減を行っている．

アップロードマネージャの管理者はストレージの全ファイルの編集が行える権限を持つこととなる．そのため，提案システムのような複数組織での使用を前提としているシステムでは，アップロードマネージャの設置場所および管理方法が問題となる．そこで本項ではアップロードマネージャの適切な設置場所・管理に関する考察を行う．

大学や企業単位などの組織での管理

大学や企業などの比較的大きな組織単位でアップロードマネージャを管理する場合について考える．前述の通り，アップロードマネージャは管理者が強い権限を持つこととなる．そのため，研究室などのさらに小さな組織単位で共同研究などを行っており，NDA などを結んでいた場合を想定するとこの管理方法は，研究室外の職員などが研究データ等の情報を取得できてしまう可能性がある．よってこの方法でのアップロードマネージャの管理は現実的ではない．しかしながら研究不正が発覚した際など，大学や企業などの監査を行うような一部の部署が研究データを見る必要がある場合が考えられる．このような場合に備え，研究グループのメンバー以外にこのような部署用の属性を作成しておき，全データのアクセス権に OR で追加するなどの工夫が必要になると思われる．

KGC を持つ組織毎の管理

提案システムの KGC を持つ組織毎にアップロードマネージャを管理する場合について考える．この方法では，第 4.2.3 項で問題となった共同研究などを行っている場合でも，

問題なくアップロードマネージャの管理が行えると考えられる．また，KGC を持つ組織毎に管理を行うため，MA-ABE において非常に強い権限を持つ KGC の管理者が既に組織内に存在すると考えられる．そのため，アップロードマネージャを管理するユーザの管理を KGC の管理者が行うことにより，実現が容易となると思われる．提案システムでのアップロードマネージャの管理に関して，KGC を持つ組織ごとに管理する方法を用いることにより，適切なアップロードマネージャの設置・運用を行うことが可能となる．また，アップロードマネージャの運用には他組織のユーザの属性の公開パラメータも必要となる．そこでアップロードマネージャは一定の周期で，全組織の全ユーザの属性の公開パラメータを取得する必要がある．この時，毎回全ての公開パラメータを取得すると非常にコストが大きくなってしまう．そのため，初回のみ全組織の全ユーザ分の公開パラメータを取得しアップロードマネージャに保存し，2 回目以降は必要に応じて定期的に新たに更新された差分の公開パラメータを取得し保存する方法を用いる．以上の方法より，提案システムにおける要件 C の「ファイルの編集・保存を権限のないユーザが実行できないこと」を満たす方法を示した．

4.2.4　各組織における属性管理の方法

提案システムは，各組織における属性の管理方法の取り決めを行う必要がある．本節ではその方法についての考察・提案を行う．提案システムに用いる Rouselakis らの方式 [9] では，安全性証明の要件を満たすため，一度 Global Setup を行い属性集合 U の領域を定義した後に，新たに属性集合 U に属性を追加することはできない．すなわち，提案システムの利用を一度始めた場合，後から属性が増えた場合でもシステムの再セットアップをしない限り，属性を追加することをできないことを意味している．しかし，実際の利用シーンを想定した場合，会社の部署や役職の新設や，大学などの改組などにおける新しい属性の追加が必要となる場合が考えられる．そこで提案システムにおいて，システム利用開始後に新たな属性の追加が必要となった場合でも対応可能な，属性の管理方法に関して考察を行う．

提案システムにおける上記の属性管理の問題に関しては，事前にある程度の大きさの属性集合の領域を定義してシステムのセットアップを行うことにより，対応することが可能と考えられる．しかしながら，実際の利用シーンでは，システムの利用開始後に追加が必要となる属性の名前である，新設される部署名や学部学科名，研究室名などは新設されて

からでないと分からないことが想定される．そのため，事前に属性を用意して管理する場合に，属性名の取り扱いが問題となる．そこで，今回は 2 つの方法を考案し，実際のシステムでの利用に適した方法についての考察・提案を行う．

属性名をマッピングテーブルで管理する方法

属性名を番号などの ID で管理を行い，ID に対応する属性名をマッピングテーブル用いて管理を行う方法が考えられる．この方法では最初の属性集合 U の領域の大きさを決める際に，属性を番号などの ID とみなして領域の確保を行う．その後，確保した領域の属性の ID に実際に使用する属性の属性名を対応させる．属性 ID と属性名の関係に関しては，図 4.3 のような形で，マッピングテーブルを用いて管理を行う．この方法をとることにより，提案システムを利用開始した後に属性を追加する必要が生じた場合でも，使用していない属性 ID に新たな属性名を割り当てることにより対応することが可能となる．

この方法では属性を追加することが可能ではあるが，属性を追加する度にマッピングテーブルの更新を行う必要がある．さらに更新を行ったマッピングテーブルは，提案システムを利用する全組織で共有をする必要があるため，更新する毎に新しいマッピングテーブルを全組織に共有しなくてはならず，非常にコストがかかるといったデメリットが考えられる．

属性名にハッシュ値を用いる方法

属性名をハッシュ関数に入力し，出力されたそのハッシュ値を属性とみなして属性集合 U で領域を確保する方法が考えられる．この方法ではまず KGC 番号と属性名を連結し（例：学生@東海大学），この連結した属性名をハッシュ関数に入力し，出力されたハッシュ値を一つの属性とみなして属性集合 U の要素の一つとして扱う．この時全ての KGC 番号において，提案システム上に存在する全ユーザの属性と連結をしてハッシュ値を取る．また KGC 番号は実際に使用する数よりも多く確保する必要がある．これにより，学部学科が増えたりした場合にも未使用の KGC 番号を割り当てることにより対応することが可能となる．提案システムにはこの方法による属性管理が適しており，この方法によって複数組織間での属性管理を行う事が可能となる．以上の内容より，提案システムにおける要件 D の「新規属性の鍵発行ができること」を満たす方法を示した．

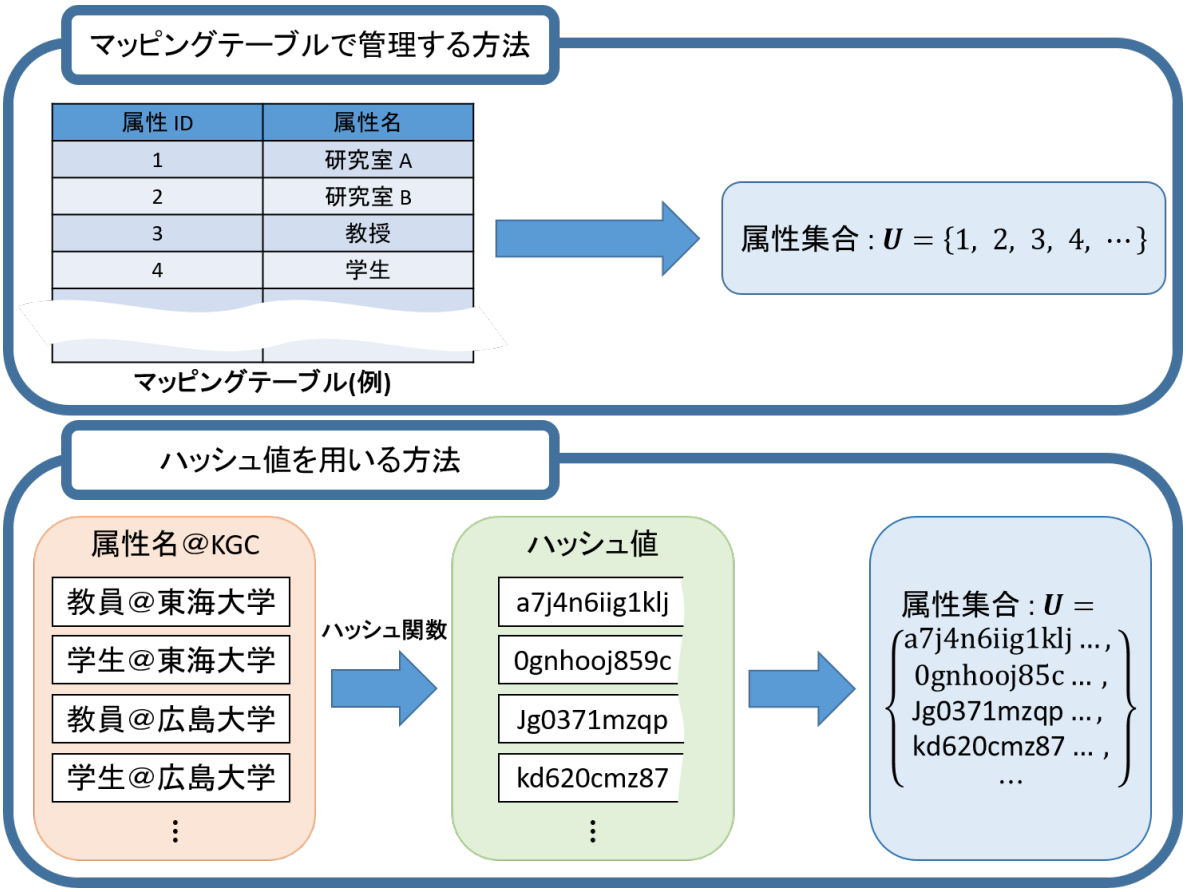

図 4.3　属性の管理方法の概要（許諾を得て文献 [3] より転載）

4.3　MA-ABE を用いたファイル共有システムの処理手順

本節では第 4.2 節での運用に関する設計・考察を踏まえ，提案システムを実際に実装・運用する際に必要となる処理手順の設計を行う．提案システムで発生する処理手順は第 4.1 節で示したように以下の項目に分類することができ，処理手順の設計をそれぞれの項目に関して行う必要がある．

- ユーザの鍵発行
- データの閲覧
- データの編集・保存

提案システムではユーザの鍵発行時に第 4.2.1 項で考察した通り，学認を用いることにより本人認証などを行った後に鍵発行を行う．第 4.3.1 項にてユーザの鍵発行に関する詳

細な処理手順を示す．次にデータの閲覧時の処理に関して，第 4.2.2 項で設計を行ったリストファイルを用いる部分の処理を含めた処理手順を第 4.3.2 項にて示す．最後にデータの編集・保存時の処理に関して，第 4.2.3 項にて設計を行ったアップロードマネージャを用いる部分の処理を含めた処理手順を第 4.3.3 項にて示す．

4.3.1 鍵発行時の処理手順

図 4.4 にユーザの鍵発行時の処理手順を示す．鍵発行を行う KGC に関しては第 4.2.1 項での説明通り，学認の仕組みを用いて SP として管理を行う．以下に鍵発行時の処理手順の説明を示す．

（K-1）本人認証

KGC 管理者はユーザに対して，事務手続きなどによる本人確認を行う．

（K-2）ePPN と認証ユーザ紐づけ

KGC 管理者は本人認証を行ったユーザとそのユーザの ePPN の紐づけを行う．

（K-3）ePPN と対応属性のデータベース登録

KGC 管理者はユーザと紐づけを行った ePPN に対して，その ePPN がどの属性を有するかを判断し，ePPN に対応する属性を KGC のデータベースに登録する．

（K-4）鍵発行要求

ユーザは自組織の KGC に対して，自身の秘密鍵を取得するため鍵発行要求を行う．

（K-5）リダイレクト

KGC は鍵発行要求を受けたユーザの認証を行うため，認証プロバイダである IdP にリダイレクトを行う．

（K-6）ユーザ認証

ユーザは IdP の認証画面にて認証を行う．

（K-7）認証情報発行

IdP は認証完了後，認証を行ったユーザの認証情報を発行する．この時の認証情報には ePPN が含まれる．

（K-8）鍵発行

KGC は IdP によって発行された認証情報を基づき，ユーザの秘密鍵を生成する．その後，生成を行った秘密鍵をユーザへ発行する．

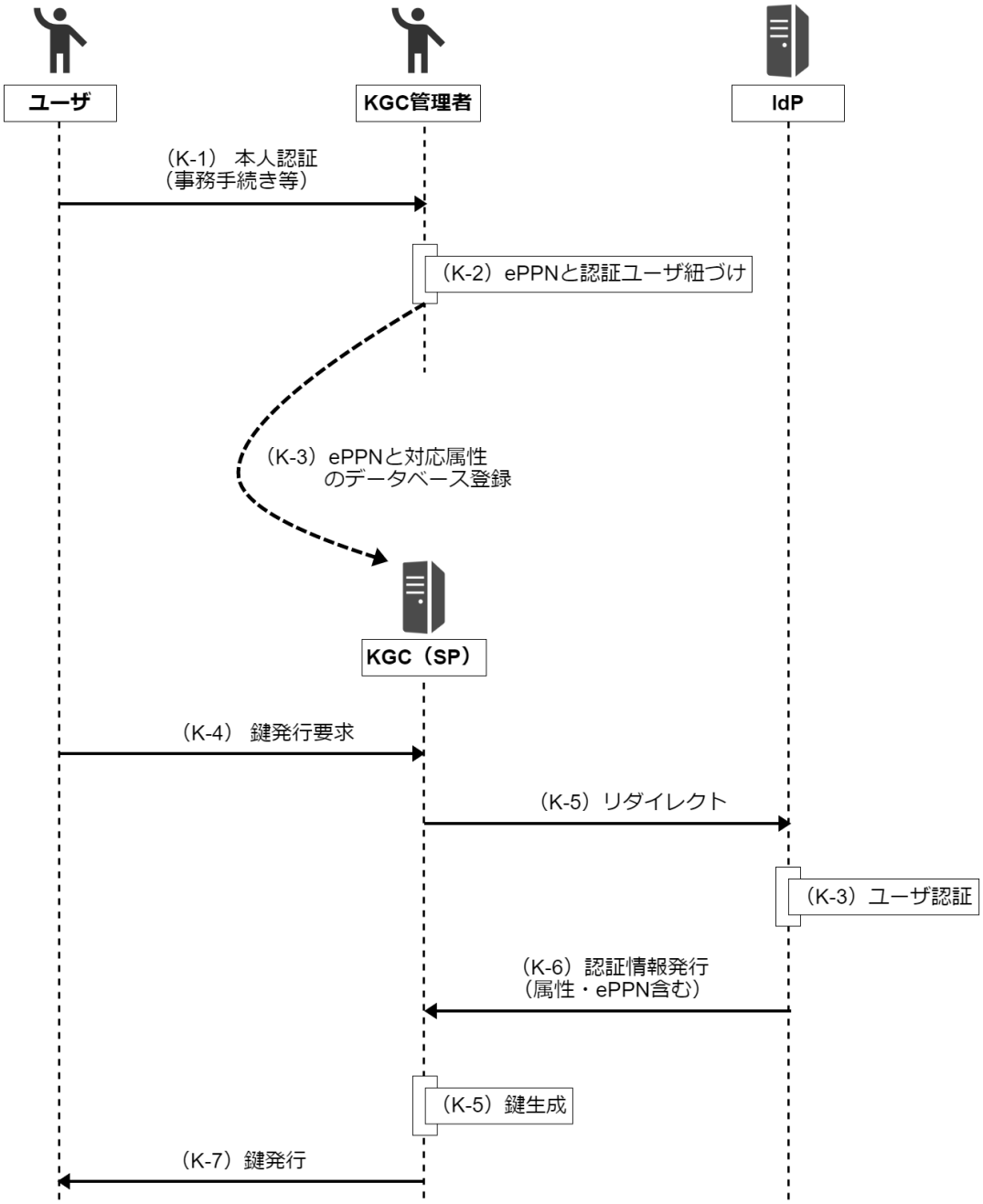

図 4.4　鍵発行時の処理手順（許諾を得て文献 [3] より転載）

ユーザは他組織においても属性を保有している場合は，別途同様の手順を所属している全組織において行う．上記の手順では，KGC の管理者が手動でユーザの本人認証と ePPN の紐づけを行う必要があるため，KGC の管理者には大きなコストがかかってしまうこととなる．そのため，本項の処理手順において KGC の管理者が手動で行う処理に関して自動化を行うことによって，更に理想的なシステムとなることが予想される．自動化を行うための方法や処理手順の設計が今後の課題である．

4.3.2 ダウンロード時の処理手順

図 4.5 にデータのダウンロード時の処理手順を示し，以下にその説明を示す．

（D-1）ディレクトリ選択

ユーザはデータをダウンロードするストレージのディレクトリ名をビューア上で確認し，移動先ディレクトリを選択する．

（D-2）リストファイル取得

選択したディレクトリのリストファイルをストレージから取得する．

（D-3）リストファイル復号

取得したリストファイルをユーザの秘密鍵にて復号する．

（D-4）ファイル名・ディレクトリ名表示

リストファイル復号後,「ファイル名・ディレクトリ名」「そのファイル・ディレクトリの read 権・write 権」「ファイル名・ディレクトリ名」を表示する．

（D-5）データ選択

ビューアに表示されているファイル名を選択する．

（D-6）データ取得

選択したデータをストレージから取得する．

（D-7）データ復号

取得したデータをユーザの秘密鍵で復号する．

リストファイルは CP-ABE を用いた従来システムを参考にしているため，ルートリストファイルを起点として，ディレクトリ構造を相対パスで保持するようになっており，（D-1）～（D-4）によるディレクトリの移動を繰り返すことにより目的のデータがあるディレクトリに到達する．

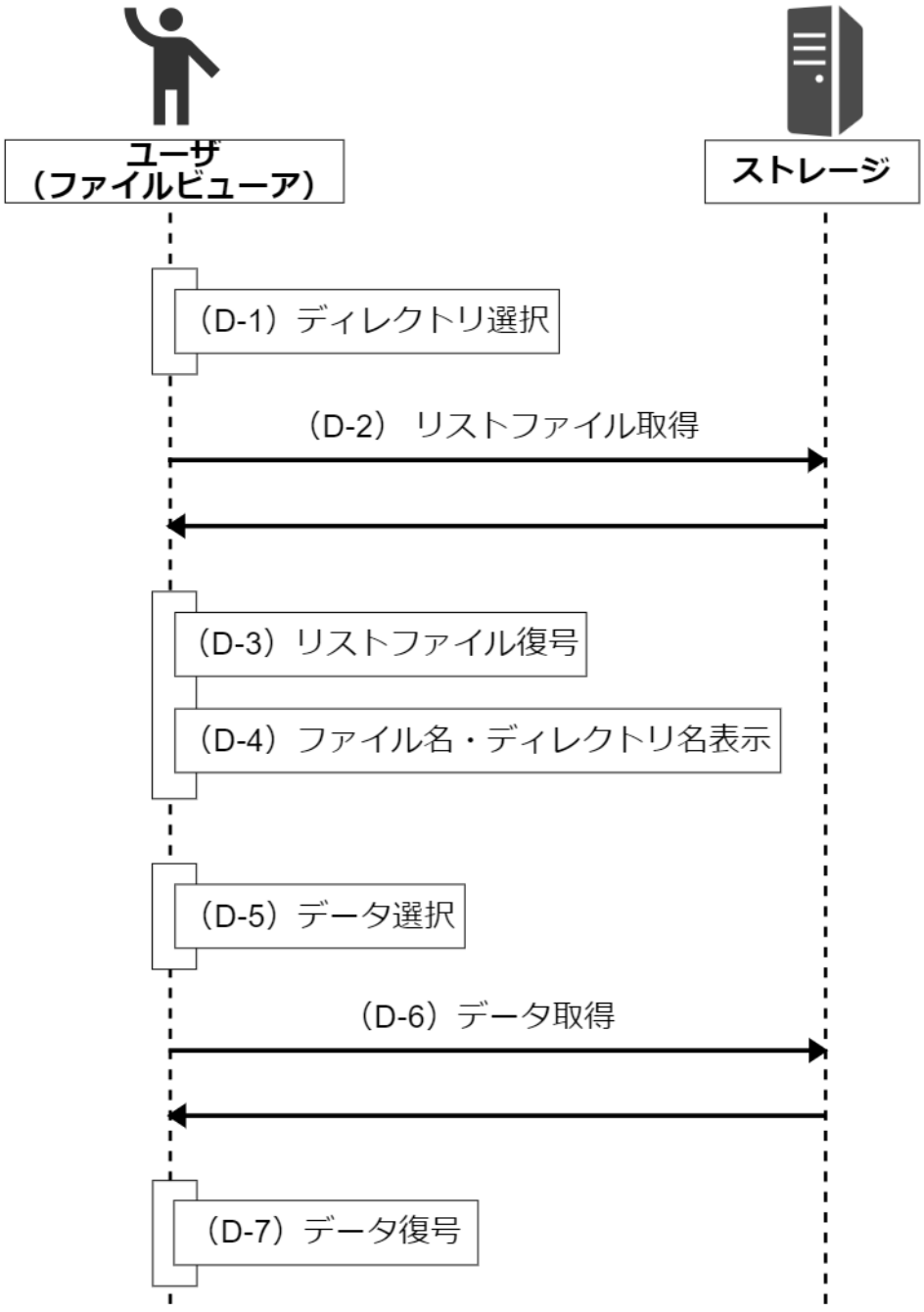

図 4.5　ダウンロード時の処理手順（許諾を得て文献 [3] より転載）

4.3.3 アップロード時の処理手順

図 4.6 にデータのアップロード時の処理手順を示し，以下にその説明を示す．

（U-1）ディレクトリの移動

ダウンロード時と同様，この手順をファイルをアップロードするディレクトリに到達するまで，この手順を繰り返す．

（U-2）ビューア操作

ユーザはビューア上にて「ファイルのアップロード」を選択し，「アップロードするファイルの選択」および「データの read 権・write 権の選択」を行う．選択操作完了後，アップロードマネージャに「ファイルのアップロード要求」が通知される．

（U-3）ユーザ認証

アップロードマネージャはチャレンジをユーザに送信し，ユーザはそれに対して write 権に対応する秘密鍵で署名を行い，その署名文を返す．この処理により，ユーザが write 権を満たす属性を有しているかを確認する．同時に「操作を行うディレクトリのリストファイルの URI」および（U-2）で決定した「データの read 権・write 権」「ファイル名の平文」を送信する．

（U-4）リストファイル取得

ストレージからリストファイルを取得し，リストファイルをロックする．

（U-5）write 権チェック

リストファイルのヘッダを参照し，ユーザが提示した write 権で書き込むことが可能か否かをチェックする．

（U-6）リストファイル復号

ヘッダを参照し，ユーザの read 権と一致するブロックの復号を行う．

（U-7）平文ファイル名重複チェック

復号したブロックの中に重複するファイル名がないかチェックする．

（U-8）保存用ファイル名生成

保存用ファイル名として疑似乱数を生成する．保存用ファイル名がストレージ上に存在しないことを確認し，予約処理としてサイズが小さいダミーファイルに保存用ファイル名を付けてアップロードしておく．既にストレージ上にファイルが存在し

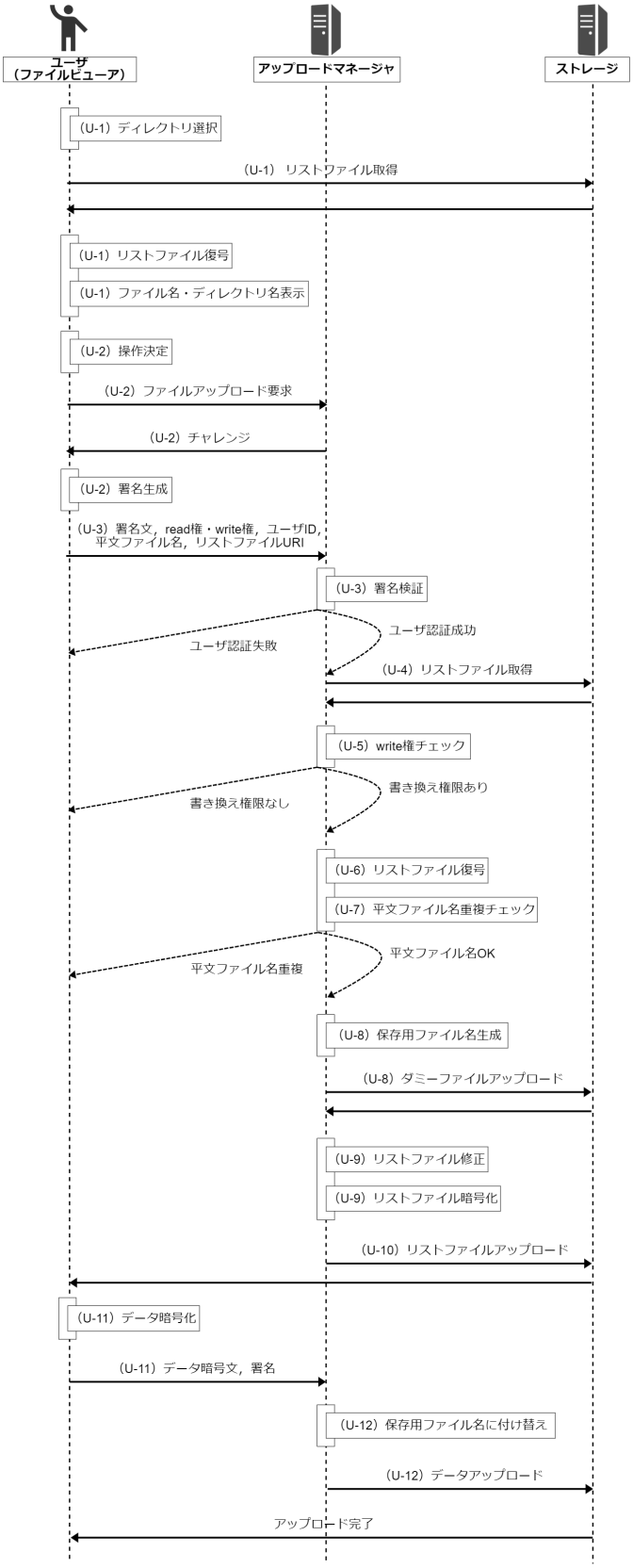

図 4.6　アップロード時の処理手順（許諾を得て文献 [3] より転載）

ていれば，保存用ファイル名の生成からやり直す．

（U-9）リストファイル修正

アップロードするファイルに関する行をブロックに追記してブロックを暗号化する．ブロックサイズ変更に伴うヘッダ情報の修正を行う．

（U-10）リストファイルアップロード

リストファイルをストレージにアップロードする．アップロード完了後にリストファイルのロックを解除する．

（U-11）データ暗号化

ビューアでデータの暗号化を行う．

（U-12）データアップロード

アップロードマネージャは暗号化したデータを保存用ファイル名に付け替え，ストレージにアップロードしてダミーファイルを上書きする．

（U-9）のリストファイル修正時に暗号化ブロックを追加する場合，その属性の公開鍵が必要となる．そのため，アップロードマネージャは全組織の全属性の公開鍵を一定の周期で取得を行い，あらかじめ保存しておく．この時，毎回全ての公開鍵を取得すると非常にコストが大きくなってしまうため，システム立ち上げ時のみ全組織の全属性分の公開鍵を取得しアップロードマネージャに保存する．2 回目以降は必要に応じて新たに更新された差分の公開鍵を取得し保存する．

4.4　実際のユースケースを想定した動作の検証

第 4 章ではここまでに，提案システムに関する詳細設計・考察を行ってきた．本節では，第 4.3 節で詳細設計を行った処理手順に，実際のユースケースを想定した処理を行った場合，第 4.2 節で挙げた提案システムが満たすべき要件をそれぞれ満たしているのか検証を行う．今回の検証を行うユーザとして，「東海大学」の「総合理工学研究科」の「学生」のユーザと「広島大学」の「情報メディア教育研究センター」の「教授」の 2 人のユーザを想定する．便宜上，本節ではこのユーザをそれぞれ「ユーザ A」「ユーザ B」と呼ぶ第 4.3 節で設計を行った 3 つの処理動作を行う場合について，それぞれの項目に分け検証を行う．

ユーザの鍵発行

まず初めに，ユーザ A，B はそれぞれこのシステムを利用するにあたって，自身の属性情報が含まれた鍵を KGC にて発行し受け取る．そのため，自組織内の KGC の管理者によって本人確認を行ってもらい，属性を割り当ててもらう．ユーザ A の場合，「東海大学」「総合理工学研究科」「学生（2023 年度）」という属性が登録されることとなり，ユーザ B の場合，「広島大学」の「情報メディア教育研究センター」の「教授（2023 年度）」という属性が登録される．この時，事前に学認 IdP にて発行される ePPN(GID) をユーザ A に紐づける形で登録することにより，各々の組織の KGC のデータベースにユーザ A，B の属性情報などがそれぞれ登録される（K-1）～（K-3）．例として，ユーザ A がすでに「学生（2022 年度）」などの古い属性情報の鍵を持っており，進学等で「学生（2023 年度）」として在籍するため，新規に「学生（2023 年度）」の属性を作成する必要がある場合がある．このような場合，システムセットアップ時に確保してある，未使用の属性空間を「学生（2023 年度）」に割り当てることとなる．また，「学生（2023 年度）」の鍵が発行された後に保存されるデータは「学生（2023 年度）」を使用して暗号化される．そのため，「学生（学生 2022 年度）」を最後に卒業するユーザなどは実質鍵が失効されたこととなり，「学生（2023 年度）」以降のデータは閲覧ができなくなる．これはユーザ B においても同様である．次に，ユーザ A，B は自身の鍵を KGC にて発行するための要求を行う．このとき，学認 IdP にてユーザの認証を行いそこから発行される ePPN(GID) などの認証情報を用いて，事前に自組織の KGC のデータベースに登録されている情報と照らし合わせ，ユーザ A，B の認証情報が合致することにより鍵が発行される（K-4）～（K-8）．以上より，ユーザ A，B は自身の属性情報に見合った鍵発行が滞りなく行われる．GID として，学認の ePPN を用いることにより全組織の全ユーザが固有の値を使用することが可能になっている．そのため，要件 1「複数組織間で利用可能なファイル共有システムであること」を満たすために必要となる，要件 A「GID は全組織の全ユーザで固有であること」が満たされ，MA-ABE を用いた際の GID の割り当てを含む鍵発行が円滑に実行されることが示された．また，ユーザは自身の属性の鍵のみを保有するだけで提案システムを使用することができるため，要件 3「ユーザの鍵管理が容易であること」を満たす．さらに，要件 D「新規属性の鍵発行ができること」が満たされることにより，要件 4「ファイルを共有するユーザの指定を動的に変更できること」を満たすことも確認できた．

データの閲覧

次にユーザ A，B は発行された鍵を用いて，提案システム上に保存されているファイルの閲覧を行う．まず初めに，ユーザは提案システムのストレージ内の情報をビューアから閲覧することにより，目的のファイルを探す（D-1）．このとき，ビューア内にてリストファイルの取得が適宜行われ，ユーザ A，B は自身の秘密鍵にて復号を行った結果，復号可能な範囲のファイル・ディレクトリ名のみが閲覧できる（D-2）～（D-4）．ユーザ A は表示されるファイルの中から「東海大学 and（教職員 or 学生）」という属性のユーザが閲覧可能である「東海大学からのお知らせ」というファイルと，「（東海大学 and（総合理工学部 or 情報通信学部））or（広島大学 and 情報メディア教育研究センター and（教授 or 准教授））」という属性のユーザが閲覧可能である「大学間交流会のお知らせ」というファイルを選択し，ダウンロードして閲覧する．「東海大学からのお知らせ」は規定されている属性情報から，「東海大学の教職員」または「東海大学の学生」という属性をもっているユーザが見ることが可能なファイルとなっているため，ユーザ A は自身の鍵で復号し閲覧することが可能である（D-5）～（D-7）．また，「大学間交流会のお知らせ」というファイルに関しても，閲覧権限を満たす属性を有するため閲覧できる．ユーザ B がこのストレージにアクセスした場合，「東海大学からのお知らせ」というファイルに規定されている属性を持ち合わせていないため，ファイル名・ディレクトリ名の閲覧をすることができない．しかし，「大学間交流会のお知らせ」に関しては閲覧権限を満たす属性を有しているため閲覧が可能となる．以上より，ユーザ A，B の属性情報に見合ったデータの閲覧が滞りなく行われた．リストファイルによって，自身の属性とは関係のないファイル名・ディレクトリ名は閲覧できていないため，要件 B「ファイル名・ディレクトリ名を含めた暗号化ができること」が満たされることが示された．よって，要件 2「共有元のユーザが指定した相手のみがファイルを編集・閲覧でき，他者にはファイルの内容がファイル名・ディレクトリ名も含めて漏洩しないこと」における，閲覧部分に関しての要件が満たされている．

データの編集・保存

最後に，ユーザ A，B は提案システム上のデータの編集・保存を行う．まず初めに，ユーザ A は，提案システムのストレージ内の情報をビューアから閲覧することにより，目的のディレクトリを探す．この時，ビューア内にてリストファイルの取得が適宜行われ，

ユーザ A の秘密鍵にて復号を行った結果，ユーザ A の鍵で復号可能な範囲のファイル名・ディレクトリ名のみ閲覧できている（U-1）．次に，ユーザ A はビューア上にてファイルのアップロードを選択し，アップロードするファイルとそのファイルの read 権・write 権を選択する．操作終了後，ユーザ A が write 権を有している場合にはデータのアップロードが行われる．この時，アップロードマネージャにアップロードが要求されユーザ A が wirte 権を有していて，そのファイル・ディレクトリの編集・削除等が可能であるかの確認処理などが行われる（U-2）～（U-5）．その他にも，ファイル名の重複やリストファイルの修正が実行され，最終的にユーザ A によってファイルがアップロードされる（U-6）～（U-12）．以上より，ユーザ A は自身の属性で編集・保存が可能なファイルをアップロードできることが示された．この時，アップロードマネージャによって権限の確認が行われ，ユーザ A が権限を有するファイルの編集・保存が行われた．ユーザ B の場合も，上記と同様の処理手順によって適切な処理が行われる．以上の結果より，要件 C「ファイルの編集・保存を権限のないユーザが実行できないこと」が満たされた．そのため，要件 2「共有元のユーザが指定した相手のみがファイルを編集・閲覧でき，他者にはファイルの内容がファイル名・ディレクトリ名も含めて漏洩しないこと」における，編集部分に関する要件が満たされることが確認できた．

上記の結果より，実際のユーザが提案システムを使用した場合において，第 4.3 節で詳細設計を行った処理手順において処理が行われることにより，要件 1「複数組織間で利用可能なファイル共有システムであること」が満たされたシステムとして動作することが示された．したがって，第 4.2 節で定義した要件 A〜D および，第 2.5.3 項で定義した要件 1〜4 を全て満たしていることが示された．

4.5　他の MA-ABE を用いる場合の考察

本章では，現在提案されている複数の MA-ABE の方式の中から，第 3 章にて選定を行った結果，提案システムに一番適していることが分かった Rouselakis らの方式 [9] を使用することを前提に設計を行った．Rouselakis らの方式は現時点で提案システムに一番適しているが，今後 Rouselakis らの方式に対する攻撃方法が発見される可能性や，より優れた MA-ABE の方式が提案される可能性がある．本節は上記のような場合に，本章で設計を行ったファイル共有システムに他の MA-ABE を用いることが可能であるかという点に関して考察を行う．

第 3.4 節にて MA-ABE の各方式を選定した際の条件について再度確認する．選定における条件は以下の 4 項目であった．4 項目それぞれについて Rouselakis らの方式と違う条件である方式について考察を行う．

- **中央機関は必要か？**
- **楕円曲線の位数**
- **安全性**
- **公開パラメータのサイズは属性数に比例せず一定か？**

初めに，Rouselakis らの方式が中央機関が必要としなかったことに対して，中央機関が必要となる方式について考察を行う．この場合，本研究の提案システムにおける複数組織間での分散管理において，中央機関を管理する組織が必要となってしまう．中央機関を管理する組織では，全組織の秘密鍵を生成できる強い権限を有するため，研究データが覗き見られる可能性が生じる．したがって，他の MA-ABE の方式を提案システムに用いる場合，中央機関は不必要な方式である必要がある．

次に，Rouselakis らの方式では楕円曲線の位数は素数位数（prime order）を用いるのに対し，合成数位数（composite order）などの他の位数を用いる方式について考察を行う．素数位数に比べて，合成数位数を用いている MA-ABE の方式では計算量が多く，パラメータサイズも大きくなってしまう．そのため，本研究では素数位数を用いた方式でのシステム提案を行った．しかし，合成数位数を用いた MA-ABE の方式において，計算量が少なく計算時間が現実的な範囲で収まり，パラメータサイズも提案システムに用いることが可能なサイズまで小さくなるような提案が行われた場合には，使用することが可能となる可能性が考えられる．そのような場合には，本研究と同様の処理時間の評価やパラメータサイズの試算等を行った上で，提案システムに用いることが可能か検討する必要があると考えられる．

その次に，Rouselakis らの方式は Selective 安全性が実現されていたのに対して，他の安全性を実現している方式について考察を行う．第 3.4.2 項にて説明した通り，Selective 安全性よりも強い Adaptive 安全性で証明されている方式である場合，処理時間が著しく遅いなど，提案システムに用いる際に他に問題となる点がない場合は，提案システムに用いることが可能であると考えられる．しかしながら，Selective 安全性よりも弱い安全性しか実現していない方式である場合には，実システムとして使用する場合に，安全に使用す

ることが可能であるかを調査する必要が生じる．

最後に，Rouselakis らの方式では公開パラメータの数が一定であったのに対して，一定とならない方式についての考察を行う．公開パラメータの数が一定とならない方式の場合，パラメータの数がどのような要素によって増減するかが重要となる．第 3.5.4 項で行ったものと同様にパラメータサイズの試算を行い，提案システムに用いることが可能なパラメータ数で収まる方式では，使用可能となる場合が想定される．しかしこの時，パラメータサイズの試算の他に公開パラメータの数が影響し，著しく処理時間が低下しないかなど，他の要素に関しても考慮する必要が生じる．

本節の結果より，本研究の提案システムに使用していた Rouselakis らの方式だけでなく，他の MA-ABE の方式を提案システムに用いることが，いくつかの条件を満たす場合に可能となることが示された．

4.6　結言

本章では，複数組織対応属性ベース暗号（MA-ABE）を用いたファイル共有システムに関して，具体的な運用方法や運用時に問題となる点を考慮した上での設計・提案を行った．まず初めに，複数組織間で利用可能なファイル共有システムの提案に基づき，提案システムを実際に運用する際には，第 2.5.3 項で定義した要件 1〜4 を満たすため，「KGC の運用方法」，「リストファイルを用いたファイル名・ディレクトリ名の管理方法」，「アップロードマネージャの運用方法と設置方法」，「各組織における属性管理の方法」，以上の要件 A〜D に関して詳細な設計が必要であることが分かった．以上の要件に関して考察を行った結果，各組織における属性管理方法には属性名にハッシュ値を用いる方法．リストファイルを用いたファイル名・ディレクトリ名の管理に関しては CP-ABE を用いた従来システム [2] の手法を参考に，提案システムに適する形で運用する方法．またアップロードマネージャの運用方法と設置方法に関しては，KGC を持つ組織ごとに管理を行う方法．KGC の運用方法は学認上にて管理を行い，ePPN を利用する方法．以上の方法によって上記の 4 つの要件 A〜D を満たす設計を示した．

次に，以上の要件を満たした提案システムを実装・運用する際に必要となる「ユーザの鍵発行」，「データの閲覧」，「データの編集・保存」に関して詳細な処理手順の設計を行った．このとき，本章にて設計した「ユーザの鍵発行」の処理手順では，KGC の管理者が手動でユーザの本人認証情報と ePPN の紐づけを行い，ePPN に対応する属性の割り当てを

行う手順が発生する．提案システムにおいては KGC を規模の小さな組織単位に細分化して管理するため，1 つの KGC 当たりのユーザ数はそこまで大きくならないため対応できる想定となっているが，KGC の管理者に手間がかかってしまうことに変わりはない．そのため，この処理を自動化で行う方法の検討・考察は今後の課題である．

その次に，実際のユースケースを想定した動作の検証を行った．第 4.3 節にて詳細設計の提案した，「ユーザの鍵発行」「データの閲覧」「データの編集・保存」のそれぞれの処理を，実システムとして利用した場合でも，第 2.5.3 項と第 4.2 節にて挙げたシステムの要件を全て満たすことを示した．

最後に，本章で設計を行った提案システムに関して，Rouselakis らの方式以外の MA-ABE を用いる場合についての考察を行った．中央機関が不必要な MA-ABE の方式の場合や，それ以外の条件である「楕円曲線の位数」,「安全性」,「公開パラメータのサイズ」が Rouselakis らの方式と異なる場合であっても，処理時間が現実的となる場合や安全性が実用上問題ない場合などのいくつかの条件を満たす場合には，提案システムに用いることが可能であることを示した．

本章により，実運用を十分に考慮した複数組織間で利用可能なファイル共有システムの具体的な要件が挙げられた．それぞれの要件を満たしたシステムの詳細設計が提案されたことに加え，実運用時の視点での運用課題の抽出やシステム構築方法の考察がされ，安全かつ実用的なファイル共有システムの実運用に関する方法が明確化された．

第 5 章

結論

本論文では，複数組織対応属性ベース暗号を用いたファイル共有システムの実現に向けて，複数組織間で利用可能な属性ベース暗号（MA-ABE）に着目し，具体的なシステムの運用方法や運用時に問題となる点を考慮した上での設計・提案を行った．

第 3 章では，複数組織対応属性ベース暗号（MA-ABE）を用いたファイル共有システムの実現可能性について考察した．まず，実用的なファイル共有のユースケースを考慮した，MA-ABE を用いた複数組織間で利用可能なファイル共有システムを提案した．今回想定しているファイル共有システムに適している既存の MA-ABE 方式について調査・比較を行った．その結果，Lewko の方式 [10] と Rouselakis らの方式 [9] が提案システムに適していることを示した．次に，Lewko の方式と Rouselakis らの方式について公開パラメータのサイズや，処理時間について具体的な数値による比較・検討を行った．Lewko の方式と Roselakis らの方式で安全性の違いがあるが，公開パラメータのサイズ，各アルゴリズムの処理時間の比較において，Rouselakis らの方式が提案システムに適していることが示された．そのため，Rouselakis らの方式を用いた提案システムにおける詳細評価として，複数の暗号化条件による処理時間および通信処理を含めた評価を行った．一番処理回数が多いと考えられる暗号化・復号に関して，属性数毎に処理時間の変化を計測した結果，暗号化に関しては属性の個数に比例して処理時間が変化し，0.13 秒未満の処理時間が属性の個数ごとに増加していくことが示された．復号処理に関しては OR 連結では属性の個数に依らず一定の時間となることが確認でき，OR 連結の復号に関しては 0.07 秒未満で処理できることが示された．AND 連結の復号に関しては，属性の個数が増える毎に 0.07 秒程度の処理時間が増加することが示された．さらに，通信時間を含めた評価を行っ

たところ，鍵発行センターからユーザが秘密鍵を取得するまでの通信時間を含めた処理時間は単一の属性の鍵では 0.31 秒程度，暗号化データの保存/閲覧に要する時間は保存時には 0.6 秒程度，閲覧時は 0.4 秒程度であることが示された．最後に，提案ファイル共有システムを実際に運用する際に問題となる．ユーザの属性変更時の鍵失効方法についての考察を行った．鍵失効機能付き MA-ABE を使用した方法，代理人再暗号化と MA-ABE を組み合わせる方法，他のエンティティを用いる方法や，鍵の属性に有効期限を埋め込む方法について検討を行った．その結果，ユーザの属性情報に有効期限を同時に設定することにより，鍵の失効を実現する方法が最も現実的であるとの結論に至った．

第 4 章では，複数組織対応属性ベース暗号（MA-ABE）を用いたファイル共有システムに関して，具体的な運用方法や運用時に問題となる要件を満たす詳細な設計・提案を行った．まず初めに，第 3 章にて提案を行った，複数組織間で利用可能なファイル共有システムを実際に運用する際に，詳細な設計が必要となる部分ついて検討および考察をした．その結果，提案システムを実運用する際には，各組織における属性管理方法には属性名にハッシュ値を用いる方法を用いる方法が最適であることを示した．次に，システム全体の管理に関して大東らの方式 [2] の手法を参考に，リストファイルを用いたファイル名・ディレクトリ名も管理を行う方法，アップロードマネージャの運用方法と設置方法に関しては，KGC を持つ組織ごとに管理を行う方法が最適であること確認した，KGC の運用方法は学認で用いられる ePPN を利用する方法が実現可能であることを示した．その次に，ユーザの鍵発行や，データのアップロード・ダウンロード時の詳細な処理手順の設計を示し，実際のユースケースを想定した動作の検証を行った．その結果，設計した詳細な処理手順が想定通りの動作をすることを確認した．最後に，本章で設計を行った提案システムに関して，Rouselakis らの方式以外の MA-ABE を用いる場合についての考察を行った．処理時間が現実的となる場合や安全性が実用上問題ない場合などのいくつかの条件を満たす場合には，提案システムに用いることが可能であることを示した．

以上のように，本研究では MA-ABE を用いたファイル共有システムの実現を目指し，実際の運用時に必要となると考えられる問題点，具体的なユースケースにおける安全性，必要な公開鍵サイズ，処理時間・通信時間，長期運用時の属性失効問題など，多角的に分析を行い，それに対応する考察・設計を示した．本研究の結果より，実運用を十分に考慮した複数組織間で利用可能な研究データ向けファイル共有システムの実現方法が明確化された．本研究の結果として，ユーザの属性変更時の鍵失効を行う方法として，第 3 章に

て考察を行った，鍵失効機能を有する MA-ABE を提案システムに用いるための検討・評価．第 4 章にて詳細な設計を行った，ユーザの鍵発行時の KGC 管理者のコストを減らすため，ユーザの本人認証および，ePPN や属性の割り当てを自動化する方法の考案・設計が今後の展望として挙げられる．本研究で得られた知見により，今後の MA-ABE を用いたファイル共有システムの研究・開発が加速することが期待される．

謝辞

本研究の過程において，終始懇切なる御指導と御鞭撻を賜り，本論文をまとめるに際して，親身な御助言と力強い励ましを頂いた，東海大学大学院総合理工学研究科総合理工学専攻 大東俊博 教授に心より感謝を申し上げます.

本研究において，数々の有益なる御教示と貴重な御助言を賜り，ご討論いただきました広島大学学長任命補佐 相原玲二 教授に深く感謝いたします.

本研究を遂行するにあたり，数々の貴重な御教示と御助言を賜りました，日本電気株式会社 土田光 博士に深く感謝いたします.

また，本研究を遂行するにあたり，様々な御助言を賜り御討論いただきました，東海大学情報通信学部通信ネットワーク工学科 柿崎淑郎 准教授に心より感謝いたします.

更に，本研究をまとめるにあたり，数々の御教示と御助言を賜りました，東邦大学理学部情報科学科 金岡晃 教授に心より感謝いたします.

本論文の審査過程において，数々の御助言と御指導を賜りました，東海大学大学院総合理工学研究科総合理工学専攻 高山佳久 教授，山本宙 教授，村山 純一 教授，森田直樹 教授に深く感謝いたします.

5 年間の大学院生活において，本研究を遂行するにあたり，温かく支え続けてくれた大東研究室の皆様に感謝いたします.

最後に，大学生活を支えて頂き，博士課程への進学にも理解を示してくれた両親と家族に心より感謝いたします.

参考文献

[1] 石橋拓哉, 小林海, 大東俊博, 土田光, 金岡晃, 柿崎淑郎, 相原玲二. 複数組織対応属性ベース暗号を用いたファイル共有システムの実現可能性に関する考察. 情報処理学会論文誌, Vol. 64, No. 3, pp. 670–686, 2023.

[2] 大東俊博, 後藤めぐ美, 西村浩二, 相原玲二. 暗号文ポリシー属性ベース暗号を利用したファイル名暗号化ファイル共有サービスの実装と性能評価. 情報処理学会論文誌, Vol. 55, No. 3, pp. 1126–1139, 2014.

[3] 石橋拓哉, 鈴木智也, 大東俊博, 土田光, 金岡晃, 柿崎淑郎, 相原玲二. 複数組織対応属性ベース暗号を用いたファイル共有システムの設計. 東海大学紀要情報通信学部, Vol. 16, No. 1, pp. 670–686, 2024.

[4] John Bethencourt, Amit Sahai, and Brent Waters. Ciphertext-policy attribute-based encryption. In *2007 IEEE Symposium on Security and Privacy (S&P 2007), 20-23 May 2007, Oakland, California, USA*, pp. 321–334, 2007.

[5] Fangming Zhao, Takashi Nishide, and Kouichi Sakurai. Realizing fine-grained and flexible access control to outsourced data with attribute-based cryptosystems. In *Information Security Practice and Experience - 7th International Conference, ISPEC 2011, Guangzhou, China, May 30 - June 1, 2011. Proceedings*, pp. 83–97, 2011.

[6] 松本悦宜, 苦木大輔, 内田恵, 近藤伸明, 満永拓邦, 五十嵐寛, 力宗幸男. 属性ベース暗号を用いたオンラインストレージサービス用クライアントの実装評価. 信学技報, Vol. 111, No. 382, pp. 73–78, 2012.

[7] 竹尾淳, 稲吉陽一朗, 白石善明, 加藤昇平, 矢口隆明, 岩田彰ほか. Hpki 認証の特長を考慮した在宅医療介護システムにおける患者情報の開示先制御. 情報処理学会論文誌, Vol. 60, No. 6, pp. 1228–1237, 2019.

[8] Melissa Chase. Multi-authority attribute based encryption. In *Theory of Cryptogra-*

phy, 4th Theory of Cryptography Conference, TCC 2007, Amsterdam, The Netherlands, February 21-24, 2007, Proceedings, pp. 515–534, 2007.

[9] Yannis Rouselakis and Brent Waters. Efficient statically-secure large-universe multi-authority attribute-based encryption. In *Financial Cryptography and Data Security - 19th International Conference, FC 2015, San Juan, Puerto Rico, January 26-30, 2015, Revised Selected Papers*, pp. 315–332, 2015.

[10] Allison Bishop Lewko. *Functional encryption: new proof techniques and advancing capabilities*. PhD thesis, 2012.

[11] Allison Lewko and Brent Waters. Decentralizing attribute-based encryption. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pp. 568–588. Springer, 2011.

[12] Tatsuaki Okamoto and Katsuyuki Takashima. Decentralized attribute-based signatures. In *Public-Key Cryptography - PKC 2013 - 16th International Conference on Practice and Theory in Public-Key Cryptography, Nara, Japan, February 26 - March 1, 2013. Proceedings*, pp. 125–142, 2013.

[13] Pratish Datta, Ilan Komargodski, and Brent Waters. Decentralized multi-authority ABE for dnfs from LWE. In Anne Canteaut and François-Xavier Standaert, editors, *Advances in Cryptology - EUROCRYPT 2021 - 40th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Zagreb, Croatia, October 17-21, 2021, Proceedings, Part I*, Vol. 12696 of *Lecture Notes in Computer Science*, pp. 177–209. Springer, 2021.

[14] NIST-FIPS Standard. Announcing the advanced encryption standard (aes). *Federal Information Processing Standards Publication*, Vol. 197, No. 1-51, pp. 3–3, 2001.

[15] Ronald L. Rivest, Adi Shamir, and Leonard M. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Commun. ACM*, Vol. 21, No. 2, pp. 120–126, 1978.

[16] Ian Blake, Gerald Seroussi, Gadiel Seroussi, and N Smart. *Elliptic curves in cryptography*, Vol. 265. Cambridge university press, 1999.

[17] Adi Shamir. Identity-based cryptosystems and signature schemes. In *Advances in Cryptology, Proceedings of CRYPTO '84, Santa Barbara, California, USA, August 19-22,*

1984, Proceedings, pp. 47–53, 1984.

[18] Dan Boneh and Matthew K. Franklin. Identity-based encryption from the weil pairing. In *Advances in Cryptology - CRYPTO 2001, 21st Annual International Cryptology Conference, Santa Barbara, California, USA, August 19-23, 2001, Proceedings*, pp. 213–229, 2001.

[19] Amit Sahai and Brent Waters. Fuzzy identity-based encryption. In *Advances in Cryptology - EUROCRYPT 2005, 24th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Aarhus, Denmark, May 22-26, 2005, Proceedings*, pp. 457–473, 2005.

[20] Vipul Goyal, Omkant Pandey, Amit Sahai, and Brent Waters. Attribute-based encryption for fine-grained access control of encrypted data. In *Proceedings of the 13th ACM Conference on Computer and Communications Security, CCS 2006, Alexandria, VA, USA, Ioctober 30 - November 3, 2006*, pp. 89–98, 2006.

[21] Adi Shamir. How to share a secret. *Commun. ACM*, Vol. 22, No. 11, pp. 612–613, 1979.

[22] Brent Waters. Ciphertext-policy attribute-based encryption: An expressive, efficient, and provably secure realization. In Dario Catalano, Nelly Fazio, Rosario Gennaro, and Antonio Nicolosi, editors, *Public Key Cryptography - PKC 2011 - 14th International Conference on Practice and Theory in Public Key Cryptography, Taormina, Italy, March 6-9, 2011. Proceedings*, Vol. 6571 of *Lecture Notes in Computer Science*, pp. 53–70. Springer, 2011.

[23] Aurore Guillevic. Comparing the pairing efficiency over composite-order and prime-order elliptic curves. In Michael J. Jacobson Jr., Michael E. Locasto, Payman Mohassel, and Reihaneh Safavi-Naini, editors, *Applied Cryptography and Network Security - 11th International Conference, ACNS 2013, Banff, AB, Canada, June 25-28, 2013. Proceedings*, Vol. 7954 of *Lecture Notes in Computer Science*, pp. 357–372. Springer, 2013.

[24] Kai Zhang, Hui Li, Jianfeng Ma, and Ximeng Liu. Efficient large-universe multi-authority ciphertext-policy attribute-based encryption with white-box traceability. *Sci. China Inf. Sci.*, Vol. 61, No. 3, p. 32102, 2018.

[25] Marloes Venema and Greg Alpár. A bunch of broken schemes: A simple yet powerful

linear approach to analyzing security of attribute-based encryption. In Kenneth G. Paterson, editor, *Topics in Cryptology - CT-RSA 2021 - Cryptographers' Track at the RSA Conference 2021, Virtual Event, May 17-20, 2021, Proceedings*, Vol. 12704 of *Lecture Notes in Computer Science*, pp. 100–125. Springer, 2021.

[26] Amit Sahai, Hakan Seyalioglu, and Brent Waters. Dynamic credentials and ciphertext delegation for attribute-based encryption. In *CRYPTO*, Vol. 7417 of *Lecture Notes in Computer Science*, pp. 199–217. Springer, 2012.

[27] Kwangsu Lee, Seung Geol Choi, Dong Hoon Lee, Jong Hwan Park, and Moti Yung. Self-updatable encryption: Time constrained access control with hidden attributes and better efficiency. In *ASIACRYPT (1)*, Vol. 8269 of *Lecture Notes in Computer Science*, pp. 235–254. Springer, 2013.

[28] Kwangsu Lee, Dong Hoon Lee, Jong Hwan Park, and Moti Yung. CCA Security for Self-Updatable Encryption: Protecting Cloud Data When Clients Read/Write Ciphertexts. 12 2018.

[29] Nuttapong Attrapadung and Hideki Imai. Conjunctive broadcast and attribute-based encryption. In *Pairing*, Vol. 5671 of *Lecture Notes in Computer Science*, pp. 248–265. Springer, 2009.

[30] Nuttapong Attrapadung and Hideki Imai. Attribute-based encryption supporting direct/indirect revocation modes. In *IMA Int. Conf.*, Vol. 5921 of *Lecture Notes in Computer Science*, pp. 278–300. Springer, 2009.

[31] Pratish Datta, Ratna Dutta, and Sourav Mukhopadhyay. Adaptively secure unrestricted attribute-based encryption with subset difference revocation in bilinear groups of prime order. In *AFRICACRYPT*, Vol. 9646 of *Lecture Notes in Computer Science*, pp. 325–345. Springer, 2016.

[32] Máté Horváth. Attribute-based encryption optimized for cloud computing. In *SOFSEM*, Vol. 8939 of *Lecture Notes in Computer Science*, pp. 566–577. Springer, 2015.

[33] Hikaru Tsuchida, Takashi Nishide, Eiji Okamoto, and Kwangjo Kim. Revocable decentralized multi-authority functional encryption. In *INDOCRYPT*, Vol. 10095 of *Lecture Notes in Computer Science*, pp. 248–265, 2016.

[34] Kenta Nomura, Masami Mohri, Yoshiaki Shitaishi, and Masakatu Morii. Attribute revo-

cable multi-authority attribute-based encryption with forward secrecy for cloud storage. *IEICE Transactions on Information and Systems*, Vol. E100.D, No. 10, pp. 2420–2431, 2017.

[35] Dawei Li, Jie Chen, Jianwei Liu, Qianhong Wu, and Weiran Liu. Efficient cca2 secure revocable multi-authority large-universe attribute-based encryption. In Sheng Wen, Wei Wu, and Aniello Castiglione, editors, *Cyberspace Safety and Security*, pp. 103–118, Cham, 2017. Springer International Publishing.

[36] Takuya Ishibashi, Toshihiro Ohigashi, and Hikaru Tsuchida. Unbounded revocable decentralized multi-authority attribute-based encryption supporting non-monotone access structures. In *International Conference on Information Technology and Communications Security*, pp. 320–339. Springer, 2022.

[37] Dai Watanabe, Hisao Sakazaki, and Kunihiko Miyazaki. Representative system and security message transmission using re-encryption scheme based on symmetric-key cryptography. *Journal of Information Processing*, Vol. 25, pp. 67–74, 2017.

[38] 市川幸宏, 山中忠和, 松田規, 坂上勉. 失効を考慮した関数型暗号システム. SCIS2012 論文集, 3D1-3, 2012.

関連発表

学術論文

[1] 石橋 拓哉，小林 海，大東 俊博，土田 光，金岡 晃，柿崎 淑郎，相原 玲二，”複数組織対応属性ベース暗号を用いたファイル共有システムの実現可能性に関する考察,” 情報処理学会論文誌, vol.64, no.3, pp.670-686, 2023 年 3 月.

紀要

[1] 石橋 拓哉，鈴木 智也，大東 俊博，土田 光，金岡 晃，柿崎 淑郎，相原 玲二，”複数組織対応属性ベース暗号を用いたファイル共有システムの設計,” 東海大学紀要情報通信学部，vol.16, no.1, 10 pages, 2024 年 3 月.（採録決定）

国際会議

[1] Takuya Ishibashi, Toshihiro Ohigashi, Hikaru Tsuchida,“Unbounded Revocable Decentralized Multi-Authority Attribute-Based Encryption Supporting Non-Monotone Access Structures,”Proc. 15th International Conference on Security for Information Technology and Communications (SECITC 2022), online, Dec. 8-9, 2022. (Revised Selected Papers, LNCS 13809, pp. 320–339, Springer, 2023)

[2] Takuya Ishibashi, Soju Komatsu, Tomoya Suzuki, Yoshio Kakizaki, Toshihiro Ohigashi, Hikaru Tsuchida, Akira Kanaoka, and Reiji Aibara,“Implementation and Evaluation of Key Generation Center for File Sharing Service using Multi-Authority Attribute Based Encryption,”The 18th Asia Joint Conference on Information Security (AsiaJCIS 2023), Tokyo, Japan, poster session, Aug. 15-16, 2023.

受賞

[1] “学生奨励賞” 情報処理学会 IOT 研究会， 2018 年 3 月.

(受賞論文)

石橋 拓哉，鈴木 達也，伊藤 勝彦，大東 俊博，相原 玲二，”属性ベース暗号を用いたファイル共有サービスの複数組織対応に関する考察,”情報処理学会 インターネットと運用技術研究会（IOT)，vol. 2018-IOT-40，pp. 1-6, 2018 年 3 月.

[2] “学生奨励賞” 情報処理学会 インターネットと運用技術シンポジウム， 2018 年 12 月.

(受賞論文)

石橋拓哉，大東俊博，土田光，金岡晃，柿崎淑郎，相原玲二，“複数組織対応属性ベース暗号を用いたファイル共有システム,”インターネットと運用技術シンポジウム 2018 論文集 (IOTS2018)，vol. 2018， pp. 16-23, 2018 年 12 月.

[3] “学生奨励賞” 情報処理学会 IOT 研究会， 2019 年 3 月.

(受賞論文)

石橋拓哉，小林海，大東俊博，土田光，金岡晃，柿崎淑郎，相原玲二，“複数組織対応属性ベース暗号を用いたファイル共有システムの評価および考察,”報処理学会 インターネットと運用技術研究会 (IOT)，vol. 2019-IOT-44, pp. 1-8, 2019 年 3 月.

学術講演

[1] 石橋 拓哉，鈴木 達也，伊藤 勝彦，大東 俊博，相原 玲二，”属性ベース暗号を用いたファイル共有サービスの複数組織対応に関する考察,”情報処理学会 インターネットと運用技術研究会（IOT)，vol. 2018-IOT-40, pp. 1-6, 2018 年 3 月.

[2] 石橋拓哉，大東俊博，土田光，金岡晃，柿崎淑郎，相原玲二，“複数組織対応属性ベース暗号を用いたファイル共有システム,”2018 年 インターネットと運用技術シンポジウム (IOTS2018)，vol. 2018, pp. 16-23, 2018 年 12 月.

[3] 石橋拓哉，小林海，大東俊博，土田光，金岡晃，柿崎淑郎，相原玲二，“複数組織対応属性ベース暗号を用いたファイル共有システムの評価および考察,”情報処理学

会 インターネットと運用技術研究会 (IOT), vol. 2019-IOT-44, pp. 18, 2019 年 3 月.