Vol.2018-IOT-40 No.14
2018/3/5

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

石橋 拓哉[1,a)]　鈴木 達也[1]　伊藤 勝彦[1]　大東 俊博[1]　相原 玲二[2]

概要：Dropbox に代表されるオンラインストレージサービスが普及してきている．このようなサービスではストレージの管理者によりデータを覗き見られる危険性があることから，ユーザ側で暗号化してデータを保護するシステムが注目されている．大東らは暗号文ポリシー属性ベース暗号 (CP-ABE) を用いてファイル本体およびファイル名・ディレクトリ名を保護できるクライアントベースの暗号化ファイル共有サービスを提案している．しかし，これらは単一組織での使用を想定している．そこで本稿では，大東の方式を複数組織で利用可能なように拡張する方法について検討する．その結果，我々が想定している利用方法では，Lewko が提案した複数機関で使用可能な CP-ABE が適していることがわかった．さらに，Lewko の方式を実装し，処理速度を評価する．

キーワード：ファイル共有サービス，暗号文ポリシー属性ベース暗号

A Study on File Sharing Services using CP-ABE Support for Multi-Authorities

Takuya Ishibashi[1,a)]　Tatsuya Suzuki[1]　Katsuhiko Ito[1]　Toshihiro Ohigashi[1]　Reiji Aibara[2]

Abstract: A lot of online storage services, e.g. Dropbox, have been widely used. These services have a weakness, which the storage administrator can obtain contents of user's files. Hence client based encryption systems are used in order to protect user's files on online storage server. Ohigashi et al. proposed a method to protect not only content of user's file but also file name of it by using Ciphertext-Policy Attribute-Based Encryption (CP-ABE). However, their method can only be used in case of a single authority. In this paper, we study methods for file sharing services using CP-ABE support for multi-authorities, and show that Lewko's de-centralized CP-ABE is suitable for file sharing services for multi-authorities. In addition, we give the implementation, and evaluate the performance.

Keywords: File Sharing Services, Ciphertext-Policy Attribute-Based Encryption

1. はじめに

クラウド技術の普及により Dropbox[*1] に代表されるクラウド上のオンラインストレージサービスが手軽に利用できるようになった．これらのサービスは自身のファイルのバックアップ以外にファイル共有の用途にも利用できる．特に自組織にサーバを設置・管理するコストを削減できるため，組織での会議用ファイルなどの共有を目的とした利用が期待される．一方，ユーザのデータは常にオンライン上のサーバに保存されているため，データの機密性や完全性の保護が課題となる．Dropbox などのオンラインストレージサービスでは，サーバでアクセス制御や暗号化を行い，アクセス権限の無いユーザからファイルを保護している．しかしながら，この方法ではストレージサービスの管理者によるデータの覗き見を防ぐことはできない．特に，政治的な理由によりディスクの検閲を実施できる国家に設置されたサーバではその懸念は大きくなる．また，管理者のオペレーションミスでユーザのデータに誤った権限が付与されてしまい，情報漏えいが発生するような事故を防ぐこともできない．このような問題を解決する方法として，ユーザ自身がファイルを暗号化するシステムを利用し，暗号化されたファイルを共有する仕組みが注目されている．

近年，柔軟なアクセス制御が可能な公開鍵暗号方式として暗号文ポリシー属性ベース暗号 (Ciphertext-Policy

[1] 東海大学 情報通信学部
2-3-23, Takanawa, Minato-ku, Tokyo 108–8619, Japan
[2] 広島大学 情報メディア教育研究センター
1-4-2 Kagamiyama, Higashi-Hiroshima, Hiroshima 739-8511, Japan
[a)] 4bjt2223@mail.tokai-u.jp
[*1] http://www.dropbox.com/

Attribute-Based Encryption: CP-ABE) [1] 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）を公開鍵とし，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復号できなくすることで，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 (KGC) に自身の属性が含まれた秘密鍵を発行してもらい，それを適切な認証を経て取得することで閲覧権限があるファイルを復号できるようになる．KGC は全てのユーザの秘密鍵を作成できる強い権限を持っているため，利用組織内の信頼できる部署が管理することを想定する．この 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 [2], [3] が議論されている．大東らはこれを拡張し，ファイルだけではなくファイル名・ディレクトリ名およびディレクトリ構造まで閲覧権限の制御範囲として保護するシステムを提案している [4].

これらの従来研究では，単一の KGC に所属しているユーザ間でのファイル共有を目的としたシステムとして議論している．KGC は所属している全ユーザの秘密鍵の生成および暗号文の復号が可能という強い権限を有しているため，実際の運用では各組織で KGC を個別に運用して組織外への情報漏えいのリスクを低減することになる．このような場合，ファイル共有サービスは組織内のファイル共有のみに用途が限定されてしまう．しかしながら，たとえば共同研究などで組織を超えた利用者のグループでファイルを共有するような利用シーンは存在する．そこで，本研究では CP-ABE を用いたファイル共有サービスを複数組織間で相互利用可能にする拡張方法について調査および考察を行う．まず初めに，従来の CP-ABE のライブラリがハイブリッド型の暗号化を採用していることに注目し，CP-ABE を用いて保護される共通鍵暗号用の共通鍵の管理を工夫することで複数組織に対応する方法およびその問題点に関して考察する．次に，上記の単純な方式を用いた場合の問題点を解決できる方式として，複数の KGC が存在可能な属性ベース暗号を調査した．その結果，我々が想定している利用方法では，中央機関を必要とせず，既存のペアリングライブラリで実装可能な方式として Lewko が提案した方式 [5] が適していることがわかった．さらに，本稿では Lewko の方式をペアリングライブラリ PBC を用いて実装・評価し，鍵取得などの事前準備で数秒程度の処理時間，平文サイズが共通鍵程度の場合で暗号化/復号に 0.5 秒未満の処理時間で実行可能であることを示す．

2. 属性ベース暗号を用いたファイル名暗号化ファイル共有サービス

本研究は CP-ABE を用いたファイル共有サービスを複数組織対応させることを目的としていることから，本章では既存のシステムとして大東らの方式 [4] を説明する．まず初めに CP-ABE について説明し，それを用いてファイル本体・ファイル名・ディレクトリ名を暗号化して開示制御が可能なシステムを概説する．

2.1 暗号文ポリシー属性ベース暗号

CP-ABE [1] 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 [6] の一種である．属性の論理式で

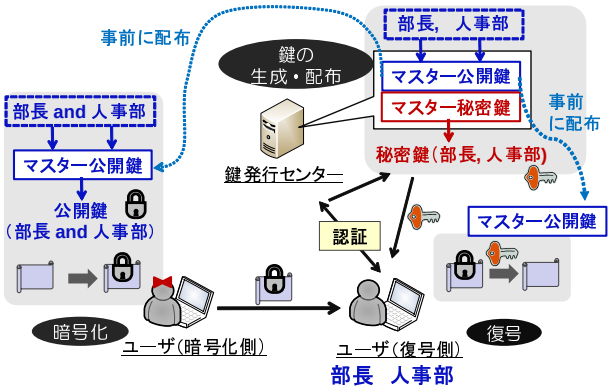

図 1 暗号文ポリシー属性ベース暗号の概要

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してもらい，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時，暗号文を復号可能となる．CP-ABE の処理の概要を図 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．

鍵発行機関 (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，鍵発行機関はユーザの 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，ユーザの秘密鍵は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可能性がある．

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型の暗号化処理が用いられることが多い．Bethencourt ら [1] が

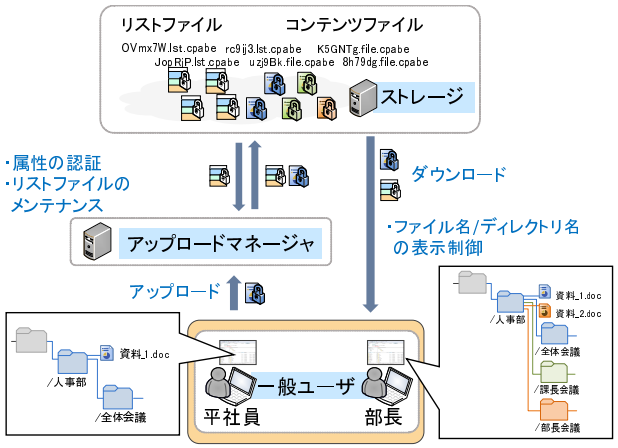

図 2　大東らの方式の概要

開発した CP-ABE のライブラリである cpabe toolkit[*2] もこのハイブリッド型の処理が実装されている．なお，このライブラリに実装されている CP-ABE のアルゴリズム [1] の詳細な計算式については紙面の都合上割愛する．

2.2　大東らの方式

本節では，大東らの方式 [4] について説明する．大東らの方式は 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．このシステムでは，ファイル名やディレクトリ名はランダムな文字列に置き換えられ，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．リストファイル内のファイル名・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 による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．なお，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，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，そこでの処理も 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．大東らの方式の概要を図 2 に示す．

3.　CP-ABE を用いたファイル共有サービスの複数組織対応

大東らの方式 [4] は単一組織内のユーザを対象にしており，共同研究や多組織が連携するプロジェクトなどでのファイル共有には向いていない．そこで，本章では CP-ABE を用いたファイル共有サービスの複数組織対応に関して，ハイブリッド型暗号化に注目した方法および複数組織を考慮した CP-ABE の二点から考察する．

3.1　ハイブリッド型暗号化に注目した素朴な方法

図 3 のようにそれぞれ組織が異なる KGC を管理している条件の下で各組織の属性を利用したアクセスポリシーを構成し，それを用いて暗号化する方法について検討する．CP-ABE で暗号化をするとき，属性の文字列だけでなく属性が所属している KGC から公開パラメータを取得する必要がある．図 3 のようなケースでは PKI（公開鍵基盤）の仕組みを利用して A 大学，B 大学，C 大学の全ての KGC の公開パラメータを安全に配布する仕組みを提供することで，別の KGC 傘下の利用者のための暗号化を実行できると思われる．しかしながら，それだけでは異なる KGC の利用者のための暗号文を作ることが可能になるだけであり，通常の CP-ABE のような AND や OR を含むような論理式を利用して柔軟に権限を指定できるわけではない．そこで，本研究ではハイブリッド型の暗号化に注目し，その暗号化手順に手を加えることで複数の組織間での CP-ABE 暗号化を実現する方法を検討する．

通常，CP-ABE を用いて巨大なファイルを暗号化する場合は処理速度の向上のためにハイブリッド型の暗号化が用いられる．ハイブリッド暗号化では，ファイル本体を暗号化している共通鍵を取り出せるかどうかを CP-ABE で制御することで属性の有無を表現している．この共通鍵を取り出す部分について工夫をすることで，属性の OR 表現や AND 表現を実現する．まず初めに属性の OR 表現について考える．複数の KGC の属性が混ざった表現で暗号化するとき，前処理として図 4 のように論理式を KGC が共通のものでまとまるように加法標準形（OR の連結）に変換する．さらに，共通鍵を含まれている KGC の数だけ複製し，それぞれを各 KGC についてアクセス権と対応する公開パラメータで暗号化する．最後にこれらの暗号文を共通鍵暗号の暗号文に連結することで，それぞれの組織のユーザは自分に関係する CP-ABE 暗号文から共通鍵を入手し，最終的に平文を得ることができるようになる．図 4 で言えば，KGC-A に属するユーザまたは KGC-C に属するユーザが共通鍵を手に入れることができ，OR 表現が実現できる．

次に同様に AND 表現について実現することを考える．これは複数の機関に所属しているユーザ (例: A 大学の教員であり，B 大学の客員研究員でもある) の権限が対応する．複数の KGC のアクセス権に対応する秘密鍵を全て持っている場合に共通鍵が入手できるようにするためには，素朴な方法として共通鍵をそれぞれのアクセス権の公開鍵で多重暗号化することが挙げられる．上記の例では，共通鍵を B 大学の客員研究員が復号できるように暗号化し，さらにその暗号文を A 大学の教員が復号できるように暗号化する．この方法で AND 表現は実現できるが，結託に関する安全性（結託耐性）に問題が生じてしまう．具体的には，アクセス権の部分木の属性を持つユーザ，たとえば A 大学の教員と B 大学の客員研究員が協力して復号処理をすることで，両方の属性を持つユーザ以外でも共通鍵を復元することができてしまう．以上のようにハイブリッド型の暗号化に注目した素朴な方式では OR 表現は可能であるが，AND 表現に結託耐性が無いことがわかる．A 大学の教員かつ B 大学の客員研究員の属性を持ったユーザだけが属する専用の KGC を用意すれば対応できる可能性もあるが，組み合わせの増加に伴って運用する KGC の数が増加すること，管理主体をどこにするかという問題もあるため，本節の方法は採用しにくい．

3.2　鍵発行機関が複数存在可能な属性ベース暗号の調査

本節では CP-ABE 自体が複数組織を考慮した方式について調査する．CP-ABE では鍵発行機関が複数存在可能な

[*2] http://acsc.cs.utexas.edu/cpabe/

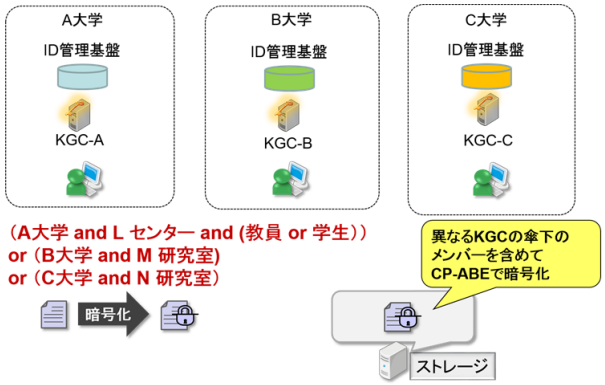

図 3　複数組織での KGC 管理の概要

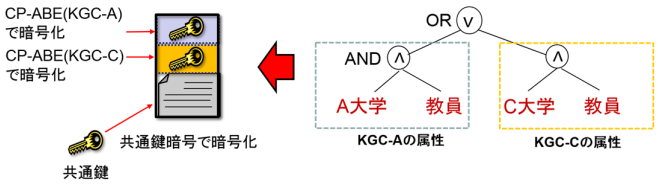

図 4　ハイブリッド型暗号での OR 表現の方法

表 1　複数の鍵発行機関が存在可能な属性ベース暗号の分類

	中央機関	楕円曲線の位数
Chase の方式 [7]	必要	Prime
Lewko らの方式 [8]	不必要	Composite
Lewko の方式 [5]	不必要	Prime
岡本らの方式 [10]	不必要	Prime
土田らの方式 [9]	不必要	Prime

方式が提案されている [5], [7], [8], [9], [10]．これらの方式は方式自体に結託耐性があるため，前節の 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 1 は，調査した方式 [5], [7], [8], [9], [10] を以下の 2 つの条件で分類したものである．

- **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るため，中央機関が不必要な方式 (De-centralized CP-ABE) の必要がある．
- **prime order の楕円曲線を使用した方式**
  複数の鍵発行機関が存在可能な属性ベース暗号は対称ペアリング暗号を用いて実現されている．そのため対称ペアリングを使用するにあたって C 言語のペアリングライブラリ，PBC library[*3] を使用している．PBC library でサポートしている楕円曲線は素数位数 (prime order) であるため，合成数位数 (composite order) の楕円曲線を必要とする方式は既存のライブラリを使用できないという観点から採用しない．

表 1 より，文献 [7] の方式は中央機関を必要としないため採用せず，文献 [8] の方式は composite order の楕円曲線を必要とするため採用しないこととした．本稿では，上記の全ての条件を満たす方式の中から Lewko の方式 [5] に注目し，実装・評価をすることにした．

[*3] https://crypto.stanford.edu

4. Lewko の方式の実装評価

本章では Lewko の方式 [5] を実装評価することで，処理時間が極端に遅くないことを確認する．

4.1 Lewko の方式

Lewko の方式では複数の KGC が独立にユーザを管理しており，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(Dual Pairing Vector Space) という技術を用いて複数の組織（KGC）が連携した暗号化を実現する．また，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．Lewko の方式のシンタックスを以下に示す．

定義 4.1　(Lewko の方式 [5])
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線形秘密分散法 (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 = Decrypt(GP, 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):　鍵発行機関 i は 12 次正方行

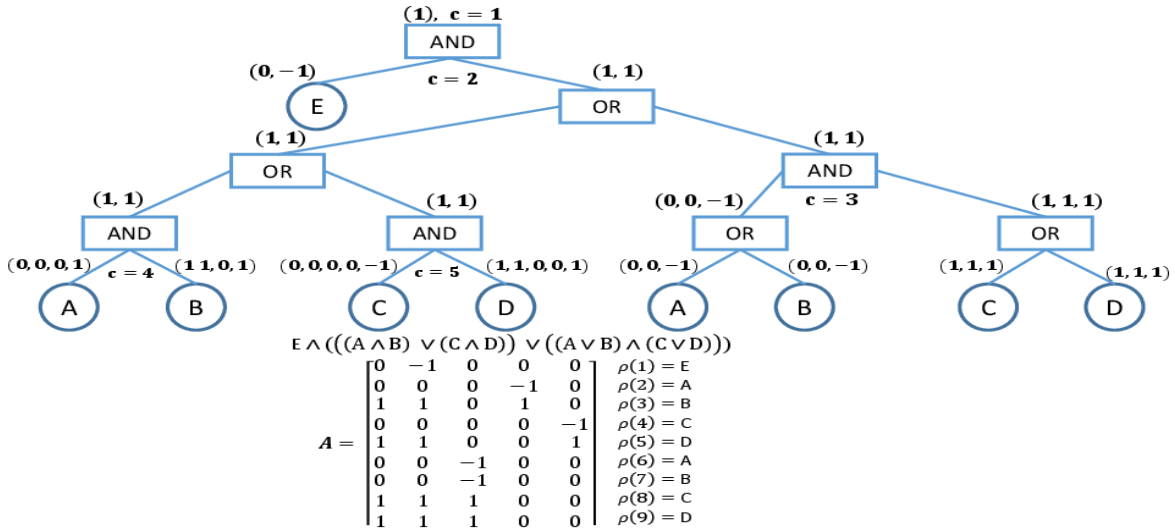

図 5 文献 [11] の線形秘密分散法の処理の概要

列 $B \in \mathbb{Z}_p^{12}$ をランダムに生成し，正規直交基底の組 $(\mathbb{B} = (\vec{b}_1, \ldots, \vec{b}_n), \mathbb{B}^* = (\vec{b}_1^*, \ldots, \vec{b}_n^*)) \in \mathbb{Z}_p^{12}$ を生成する．さらに，鍵発行機関 i は 2 つの一様乱数 ${\alpha_i}^1, {\alpha_i}^2 \in \mathbb{Z}_p$ を選択する．鍵発行機関 i は公開パラメータ $\mathrm{PK} = \{e(g,g)^{a_i^1}, e(g,g)^{a_i^2}, g^{\vec{b}_{1,i}}, g^{\vec{b}_{2,i}}, g^{\vec{b}_{3,i}}, g^{\vec{b}_{4,i}}\}$ と秘密鍵 $\mathrm{SK} = \{g^{\alpha_i^1 \vec{b}_{1,i}^*}, \vec{b}_{1,i}^*, \vec{b}_{2,i}^*, \vec{b}_{3,i}^*, g^{\alpha_i^2 \vec{b}_{3,i}^*}, \vec{b}_{4,i}^*\}$ 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．$C = Me(g,g)^s, D_j = e(g,g)^{A_j \cdot v} e(g,g)^{\alpha_{\rho(j)}^1 r_j^1} e(g,g)^{\alpha_{\rho(j)}^2 r_j^2}$，$C_j = g^{r_j^1 \vec{b}_{1,\rho(j)}} + (r_j^1 + A_j \cdot w^1)\vec{b}_{2,\rho(j)} + r_j^2 \vec{b}_{3,\rho(j)} + (r_j^2 + A_j \cdot w^2)\vec{b}_{4,\rho(j)}$ 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): 鍵発行機関 i は $H(\mathrm{GID}) = (H_{\mathrm{GID}}^1, H_{\mathrm{GID}}^2) \in \mathbb{G}$ を計算する．復号するユーザの秘密鍵 $K_{i,\mathrm{GID}} = g^{\alpha_i^1 \vec{b}_{1,i}^*} g^{\alpha_i^2 \vec{b}_{3,i}^*} (H_{\mathrm{GID}}^1)^{\vec{b}_{1,i}^* - \vec{b}_{2,i}^*} (H_{\mathrm{GID}}^2)^{\vec{b}_{3,i}^* - \vec{b}_{4,i}^*}$ 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．

4.1.1 線形秘密分散法

Lewko の方式ではアクセス権を表現する行列 A は線形秘密分散法 (LSSS) で作成する．本節では本実装で採用した 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り，Lewko の方式の **Decrypt** での $\omega_1, \ldots, \omega_j$ の選択が容易になるが詳細は省略する．この LSSS の処理の概要の例を図 5 に示す．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，親ノードの持つアクセスベクトルを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．

4.2 実装および評価

本章では Lewko の方式を実装・評価する．Lewko の方式は対称ペアリングから構成されるため，PBC Library (version 0.5.14) 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

表 2　計測に使用した機器の仕様

	PC1
CPU	Intel(R) Core(TM) i7-6950x @ 3.00GHz
Memory	64GB
OS	Linux version 3.16.1-4-amd64

表 3　計測結果

	処理時間 [sec]
Global Setup	0.096
Authority Setup	2.423
Encrypt	0.343
KeyGen	1.368
Decrypt	0.410

トで 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て 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．開発言語は C 言語を用いている．

本稿は速報の位置づけで Lewko の方式の最低限の評価のみを行うことに注意されたい．属性数や平文サイズを変化させた評価については稿を改めて実施する予定である．今回の実験では，暗号化/復号での平文 M は拡大体 $\mathbb{G}_T$ の一つの要素にエンコード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。また，実験において KGC の数は 4 とし，それぞれの KGC は 1 つの属性を有しているとして暗号化を行う条件で実験をしている．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．この条件のみでは論理演算と処理の関係を示しているとは言えないが，4 種類の KGC の属性が混合されたアクセス権での暗号化が動作すること，および処理が極端に遅くは無いことを確認できると考えている．

表 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表 3 に示す．KGC を構築したときの処理である **Authority Setup** やユーザが鍵を取得する処理である **KeyGen** は 3 秒未満で実行できている．これらは最初に一度のみ実行する処理であるため許容できる処理時間だと考える．暗号化/復号処理である **Encrypt** や **Decrypt** は 0.5 秒未満で実行できている．

5. まとめ

本稿では CP-ABE を用いたファイル共有サービスにおいて，複数組織対応のための方法について考察した．その結果，結託耐性を得るために複数の鍵発行機関が存在可能な属性ベース暗号が適していることがわかった．さらに，その中で Lewko の方式 [5] を実装し，ある条件において秘密鍵の取得など事前処理は 3 秒未満，共通鍵など小さいデータの暗号化/復号処理は 0.5 秒未満で実行できることがわかった．

今後の課題は以下の通りである．まず，アクセス権に含まれる属性数および平文のサイズを変更したときの Lewko の方式の評価が必要である．さらに，本稿で示したアルゴリズムは簡単化のために各 KGC で単一の属性を扱うようにしていたが，それを複数の属性を扱う形に表現した上で実際のシステム上で実装・評価することも課題である．また，今回は Lewko の方式を選択したが，岡本らの方式 [10] や土田らの方式 [9] も同様に目的に合致しているため，それらの評価も今後の課題としたい．

謝辞　本研究の一部は JSPS 科研費 JP15K00185, 16H02808 の助成，MIC/SCOPE (課題番号 162108102) の委託を受けたものである．

参考文献

[1] Bethencourt, J., Sahai, A. and Waters, B.: Ciphertext-Policy Attribute-Based Encryption, *2007 IEEE Symposium on Security and Privacy (S&P 2007), 20-23 May 2007, Oakland, California, USA*, pp. 321–334 (2007).

[2] Zhao, F., Nishide, T. and Sakurai, K.: Realizing Fine-Grained and Flexible Access Control to Outsourced Data with Attribute-Based Cryptosystems, *Information Security Practice and Experience - 7th International Conference, ISPEC 2011, Guangzhou, China, May 30 - June 1, 2011. Proceedings*, pp. 83–97 (2011).

[3] 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).

[4] 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).

[5] Lewko, A. B.: Functional encryption: new proof techniques and advancing capabilities, PhD Thesis (2012).

[6] Sahai, A. and Waters, B.: Fuzzy Identity-Based Encryption, *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).

[7] Chase, M.: Multi-authority Attribute Based Encryption, *Theory of Cryptography, 4th Theory of Cryptography Conference, TCC 2007, Amsterdam, The Netherlands, February 21-24, 2007, Proceedings*, pp. 515–534 (2007).

[8] Lewko, A. and Waters, B.: Decentralizing attribute-based encryption, *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, Springer, pp. 568–588 (2011).

[9] 土田　光，金山直樹，西出隆志，岡本栄司：Non-Programmable ランダムオラクルモデルで安全性証明可能かつ複数の鍵発行機関が存在可能な属性ベース暗号，信学技報，Vol. 115, No. 502, pp. 197–204 (2016).

[10] Okamoto, T. and Takashima, K.: Decentralized Attribute-Based Signatures, *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).

[11] Lewko, A. B. and Waters, B.: Decentralizing Attribute-Based Encryption, *IACR Cryptology ePrint Archive*, Vol. 2010, p. 351 (2010).