Vol.2012-IOT-16 No.36
2012/3/16

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

後藤 めぐ美[†]　　大東 俊博[††]
西村 浩二[††]　　相原 玲二[††]

近年，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，TrueCrypt のようなクライアント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．最近では，暗号文ポリシー属性ベース暗号（CP-ABE）と呼ばれる新しい公開鍵暗号の方式を利用することで，きめ細やかなアクセス制御をクライアント主導で行う方法について活発に議論されている．しかしながら，既存の方式ではファイル名やディレクトリ名の秘匿については議論されていなかった．そこで，本稿では CP-ABE と属性ベース署名を利用してコンテンツの暗号化だけでなくファイル名やディレクトリ名の暗号化および編集権限の制御が可能な方式を提案し，ファイル共有サービスへ応用する．

A File Sharing Service with File Name Encryption using CP-ABE

Megumi Goto[†], Toshihiro Ohigashi[††], Kouji Nishimura[††] and Reiji Aibara[††]

Recently, 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 like TrueCrypt are used in order to protect user's files on online storage server. In particular, several researchers discuss methods based on Ciphertext-Policy Attribute Based Encryption (CP-ABE), which encrypt data with flexible access control by client-side. However, previous methods don't discuss confidentiality of file name and directory name. In this paper, we propose a method to protect not only content of user's file but also file name of it by using CP-ABE and Attribute Based Signature (ABS). The proposed method can protect directory name too, and can control write permission of file/directory by client-side. We apply this method to file sharing service with file name encryption.

1. はじめに

クラウド技術の普及により DropBox[a)]をはじめとするオンラインストレージが手軽に利用できるようになった．クラウドのストレージサービスは自身のファイルのバックアップ以外にも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するコストを削減できるため，組織内でのファイル共有目的での利用が期待される．一方，ユーザのデータが常にオンライン上のサーバに保存されるため，データの機密性や完全性の保護が課題となる．DropBox などのストレージサービスではサーバ側でアクセス制御や暗号化を行い，アクセス権限の無い利用者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に設置されたサーバではその懸念は大きくなる．そこで，利用者が自らデータを保護する方法として，TrueCrypt[b)]などのクライアント側でコンテンツを暗号化するシステムが注目されている．しかし，これらは共通鍵暗号や従来の公開鍵暗号を利用してコンテンツを暗号化するため，ユーザはファイル共有したいグループ数の鍵を配布・管理をする必要があり，組織内でのファイル共有などの用途ではコストが大きくなる．さらに，既存のシステムの多くはコンテンツのみを暗号化の対象とし，ファイル名やディレクトリ名を保護しない．ファイル名やディレクトリ名はその内容を要約する情報が含まれている場合も多く，アクセス権限の無い利用者に知られることは望ましくない．

クラウド上のサービスに有効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 Attribute Based Encryption: CP-ABE）[1]が提案されている．CP-ABE は属性ベース暗号[2]の一方式であり，暗号文の中に属性値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）を埋め込むタイプの暗号である．利用者はアクセス権を公開鍵として利用することで復号できる利用者のグループを任意に設定できる．また，利用者が管理する秘密鍵の個数は自身の属性の個数に依存するため，組織内でのファイル共有のようなファイル共有をしたいグループ数が多くなる場合でも有効となる．近年，CP-ABE によってクライアント側でファイルを暗号化することで，クラウド上のストレージにおけるファイルの閲覧の権限を制御する研究が盛んに行われている[3][4]．さらに，Zhao らは属性ベース署名[5]を用いて利用者の属性を確

† 広島大学大学院総合科学研究科
Graduate School of Integrated Arts and Sciences, Hiroshima University
†† 広島大学情報メディア教育研究センター
Information Media Center, Hiroshima University

a) http://www.dropbox.com/
b) http://www.truecrypt.org/

認する機能をストレージサーバに組み込み，ファイルを編集可能な権限も利用者側で決定できるように拡張している．しかし，これらの方式はファイル名/ディレクトリ名の秘匿に対しては考慮されていない．

既に著者らは CP-ABE を用いてファイル名のような小さなデータを多数処理するための方式を提案している[6]．本稿では，その一方式を発展させ，CP-ABE と属性ベース署名を用いてファイル名/ディレクトリ名の表示制御および編集権限の管理が可能な方式を提案する．ファイル名/ディレクトリ名の表示制御を高速に行うため，リストファイルと呼ぶグループ単位でファイル名/ディレクトリ名をまとめて扱うファイルを導入する．さらに，ファイルの編集権限を制御するためにアップロードマネージャと呼ぶ属性ベース署名を利用した管理システムを導入する．このシステムはストレージサーバと独立に設置できるため，既存のストレージサービスに変更を加えることなく利用できる．

2. 準備

2.1 暗号文ポリシー属性ベース暗号

本節では CP-ABE の処理の概要（図 1）について述べる．CP-ABE は所属や役職などの属性を公開鍵として利用できる属性ベース暗号の一種であり，属性の論理式で表現されたアクセス権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．たとえば，人事部 OR (総務部 AND 部長)が復号できるように暗号化をした場合，総務部の部長は{総務部，部長}の属性に対応する秘密鍵を利用して復号できる．

CP-ABE は信頼できる第三者機関として鍵発行センターを置く必要がある．鍵発行センターはマスター公開鍵とマスター秘密鍵を作成し，全ユーザにマスター公開鍵を配布する．ユーザは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．また，属性に対応する秘密鍵は鍵発行センターによってマスター秘密鍵と属性値から作成される．鍵発行センターはユーザの ID と属性の対応表を保持しておき，ユーザを認証した上で対応する秘密鍵を安全に配布する．CP-ABE のアルゴリズムの詳細については紙面の都合上割愛する．なお，鍵発行センターは全属性の秘密鍵を生成できるマスター秘密鍵を持つため，全ての暗号文を復号できる強い権限を持つ．

CP-ABE は柔軟な暗号化が可能であるが公開鍵暗号であるため，AES などの共通鍵暗号と比べると低速である．そのため一般的に，サイズが比較的大きいコンテンツ本体は共通鍵暗号で暗号化し，それに用いる小さなデータである共通鍵（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処理が用いられる．文献[1]の著者らが開発した CP-ABE のライブラリ cpabe toolkit[c]もハイブリッド型の処理が実装されており，本研究ではこのライブラリを使用することを前提として議論する．

c) http://acsc.cs.utexas.edu/cpabe/

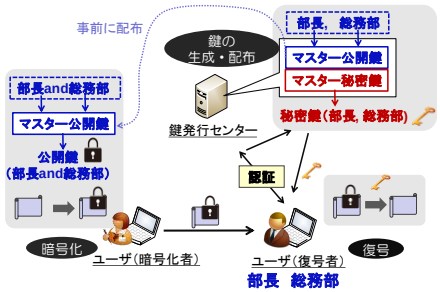

図 1　CP-ABE の暗号化/復号処理

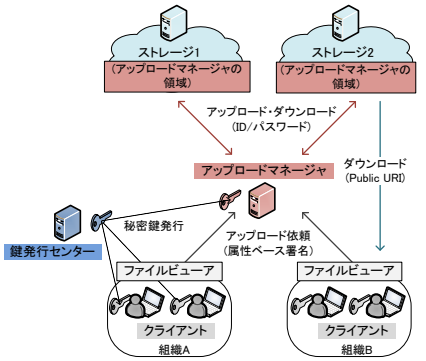

図 2　システム全体構成

2.2 属性ベース署名

属性ベース署名は属性ベース暗号の署名版であり，属性に対応した秘密鍵で署名文を作成し，属性値とマスター公開鍵を用いて署名を検証することができる．全体の構成は CP-ABE と同様であるため詳細な説明は省略する．なお，本研究では属性ベース署名を認証目的で利用している．

3. リストファイルを利用したファイル名/ディレクトリ名の表示制御方法

一般的なファイル暗号化システムではコンテンツの暗号化のみを行うが，提案システムではコンテンツだけでなく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．なぜなら，ファイル名やディレクトリ名はその内容を要約する情報が含まれている場合があるからである．本稿において，アクセス権のうちコンテンツまたはファイル名/ディレクトリ名を復号できる権限を read 権，それらを作成または編集できる権限を write 権と定義して議論をしていく．

提案方式の read 権の制御は既存の研究と同様に CP-ABE を用いて実現する．ただし，ファイル名やディレクトリ名についてはリストファイルという概念を用いて高速に処理をする．リストファイルの詳細は 3.2 節に記述している．

共有の ID を利用するなどしてファイルを共有すると，共有領域に自由にファイルを作成し参照できるが，ID を共有している他のユーザに勝手にファイルを削除されてしまう．そこで，我々はユーザの write 権をアップロードマネージャが集中管理する方法を採用している．アップロードマネージャは属性ベース署名を利用してユーザの write 権を確認し，write 権を満たすユーザの要求のみストレージへの書き込みを代行する．write 権の管理およびファイル名/ディレクトリ名の秘匿はリストファイルを利

用しアップロードマネージャがメンテナンスを行う．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分散を考慮する．（図 2）

3.1 管理者の分類と一般ユーザ

各管理者と一般ユーザの権限と役割について整理する．

ストレージ管理者　クラウドのストレージサービス提供者であり，ユーザ認証した者のみファイルのアップロードを許可する機能を一般ユーザに提供している．特に本稿では，ダウンロードを不特定多数に許可するサービスが提供されていることを想定している．たとえば DropBox では，利用者 ID を持たないユーザともファイル共有を行えるよう，パブリックリンクとよばれる読み出しのみ可能な URI を利用できる．ストレージ管理者はストレージ上のファイルを覗き見することが可能とする．

アップロードマネージャ管理者　ストレージの ID とパスワードを管理し，ストレージに書き込みのできる唯一のユーザである．ユーザの要求に応じてユーザの代わりに書き込みを行い，リストファイルのメンテナンスを行う．

鍵発行センター管理者　ユーザの ID とそれに紐づく属性を管理し，一般ユーザやアップロードマネージャに CP-ABE と属性ベース署名で利用する鍵の生成と配布を行う．ユーザの秘密鍵を生成するためのマスター秘密鍵を管理するので，全ての暗号文を復号できる強い権限を持つ．

一般ユーザ　ID と属性を持ち，自分の属性が埋め込まれた秘密鍵を鍵発行センターから入手できる．アップロードはアップロードマネージャを介し，ダウンロードはストレージから直接行う．アップロードやダウンロードの処理にはファイルビューアを利用する．

表 1　管理者および一般ユーザの権限

	コンテンツの復号	ファイル名・ディレクトリ名の復号/ディレクトリ構造の取得	ストレージ上への暗号化ファイルの作成
ストレージ管理者	×	×	×
アップロードマネージャ管理者	×	○	○
鍵発行センター管理者	○	○	×
一般ユーザ（属性合致の場合）	○	○	×

表 1 に上記のプレイヤーが持つ権限をまとめた．コンテンツ（ファイルの本文）の暗号文はコンテンツの read 権を満たす復号鍵を持つ一般ユーザもしくは鍵発行センターしか復号することができない．アップロードマネージャやストレージ管理者にはコンテンツに関する復号鍵を渡さないためコンテンツを秘匿することができる．アップロードマネージャはリストファイルの管理のため，一般ユーザは表示のため，ファイル名/ディレクトリ名に関する復号鍵を持つ．正しく暗号化したファイルを作成しストレージへアップロードする操作は，ID とパスワードが必要であるため，アップロードマネージャ管理者のみが可能である．ストレージ管理者によるファイルの修正・削除は CP-ABE など暗号化技術のみでは防ぐことはできない．しかしながら，Web 改ざんをリモートで監視するシステム[d]などを導入し，アップロードマネージャがアップロードしたファイルが不正に変更されることを監視すれば単純な置き換えによる改ざんを防ぐことは可能である．

3.2 リストファイルの役割と利用方法

通常，ファイル名やディレクトリ名は一度に複数表示するため，ファイル名やディレクトリ名の暗号化・復号をファイル/ディレクトリ個別に行うことは CP-ABE の小さな処理を多数発生させることになり処理時間が増加する．そこで，あるディレクトリにおいて，同じ read 権を持つファイル名や子ディレクトリ名を 1 つのファイル（リストファイル）で管理して高速に復号処理を行う．リストファイルは read 権で暗号化してストレージに保存する．リストファイルはディレクトリ毎に作成し，ディレクトリを移動する都度リストファイルを復号してファイル名/ディレクトリ名表示を行う．リストファイル同士は URI で相互リンクして仮想的にディレクトリ構造を作り上げる．リストファイルは，図 4 で示すようにファイルの物理的な保存場所とディレクトリの論理的な構造を結びつけている．

リストファイルは同じ read 権を持つ複数のユーザで共有するため，不正な書き換えや削除を防ぐためにユーザはアップロードマネージャを介してリストファイルの編集を行う必要がある．リストファイルの中にファイルやディレクトリの write 権を表す文字列を格納し，アップロードマネージャはこれを確認して操作する．アップロードマネージャは write 権を確認するため，ユーザに属性ベース署名を要求し write 権を満たす属性を持つユーザであるかをチェックする．アップロードマネージャはすべてのリストファイルをメンテナンスするため，リストファイルの read 権にアップロードマネージャの属性を OR で追加する必要がある（ただしコンテンツの復号権は与えない）．

ディレクトリの中には複数の read 権を持つリストファイルが存在するため read 権毎にリストファイルを管理するとディレクトリ名の変更などの際の扱いが煩雑になる．そこで，図 3 のようにディレクトリ単位で一括して扱えるように，リストファイルは read 権ごとに暗号化されたブロックと平文のヘッダ情報から構成している．各ブロックを暗号化するアクセス権はそのディレクトリの read 権を表し，ユーザは復号可能なブロックの情報のみファイル名/ディレクトリ名を表示できる．各ブロックはアップロードマネージャによりメンテナンスされるため，アップロードマネージャの属性も OR で追加したアクセス権で暗号化する．ブロックの各行は行頭の 3 種類の識別子（DP,

d) http://www.kaizankenchi.jp/

DC，F）で内容を区別し，各列はタブ区切りで次の情報を保持する．

- DP[tab]親ディレクトリのリストファイルの URI
- DC[tab]子ディレクトリ名の平文[tab]リストファイルの URI
- F[tab]カレントディレクトリに存在するコンテンツ名の平文[tab]
  コンテンツの URI[tab]ファイルの write 権[tab]ファイル作成者の ID

ヘッダ情報には，1 行目にカレントディレクトリの write 権を表す文字列及びディレクトリ作成者の ID，2 行目以降には read 権毎のブロックのリストファイル内での開始位置を表すバイト数及びそのブロックの read 権を表す文字列を格納する．

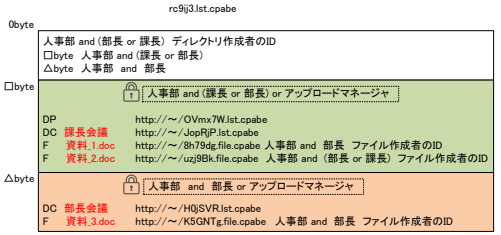

図 3　リストファイルのフォーマット

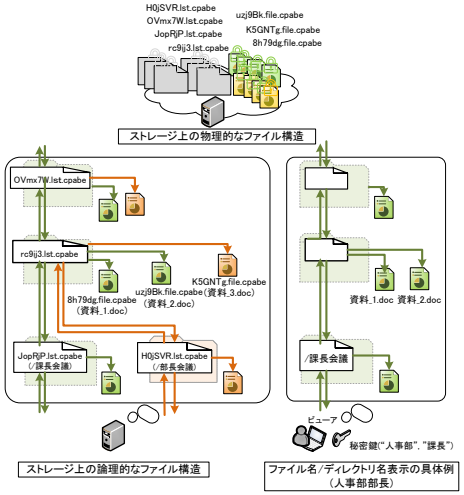

図 4　ストレージ上のファイル構造

3.3 ファイルのダウンロード処理手順

ファイルのダウンロード処理（図 5）は下記の手順で行われる．

(D-1)ディレクトリ選択
ビューアに表示されているディレクトリ名の中から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
選択され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で復号する．

(D-4)ファイル名/ディレクトリ名の表示
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/ディレクトリの作成者」を表示する．

(D-5)コンテンツ選択
ビューアに表示されているファイル名を選択する．

(D-6)コンテンツ取得
選択されたコンテンツをストレージから取得する．

(D-7)コンテンツ復号
取得したコンテンツをユーザの秘密鍵で復号する．

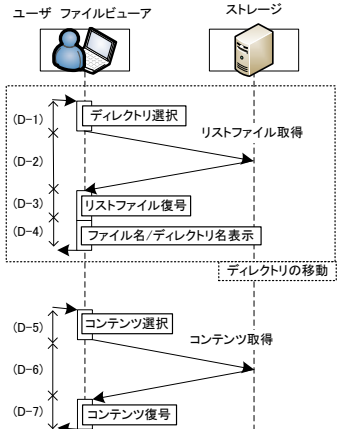

図 5　ファイルのダウンロード処理手順

ファイルのダウンロード処理について説明する．リストファイルは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されており，(D-1)～(D-4)によるディレクトリの移動を繰り返すことで目的のディレクトリに到達する．ルートリストファイル名は固定名でビューアに内蔵（preset）しておく．リストファイルの復号，コンテンツの暗号化および復号するための秘密鍵は，ビューア起動時にユーザ認証して鍵発行センターから取得する．

3.4 ファイルのアップロード処理手順

ファイルのアップロード処理（図 6）は下記の手順で行われる．

(U-1)ディレクトリの移動
ファイルのダウンロード処理手順(D-1)～(D-4)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繰り返す．

(U-2)操作決定
ユーザはビューアで「ファイルのアップロード」操作を選択し，「アップロードするファイルの選択」「コンテンツの read 権・write 権の選択」を行う．選択完了後，アップロードマネージャに「ファイルアップロード要求」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 権および ID に対応する秘密鍵で署名をし，その署名文を応答する．署名文と同時に「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「コンテンツの read 権・write 権」「ファイル名平文」「ユーザの ID」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
リストファイルヘッダを参照して，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
ヘッダ情報を参照し，ユーザの read 権と一致するブロックを復号する．

(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)保存用ファイル名重複チェック
保存用ファイル名を生成する（擬似乱数生成）．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た上で，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（予約処理）．もし既にストレージ上にファイルが存在していれば，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)コンテンツ暗号化
ビューアでコンテンツの暗号化を行う．

(U-12)コンテンツアップロード
アップロードマネージャは暗号化したコンテンツ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．

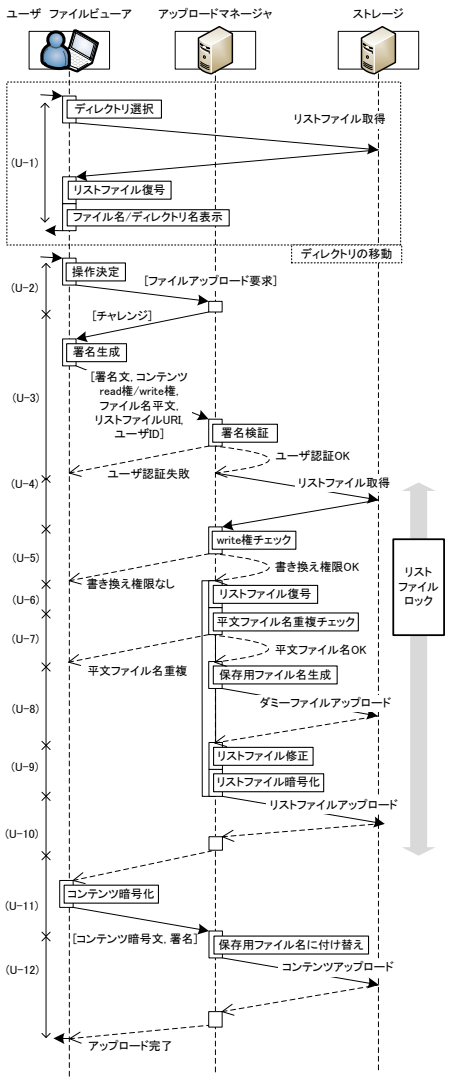

図 6 ファイルのアップロード処理手順

ファイルのアップロード処理について説明する．(U-2)でコンテンツの read 権や write 権を決定するときはチェックボックスなどの GUI を利用して選択する．コンテンツの read 権や write 権は，アップロードするディレクトリの write 権をデフォルトとし，変更も可能とする．(U-3)で生成した署名文は(U-11)でも利用し，ビューアとアップロードマネージャの通信が同一セッションであることを確認する．ブロックが異なる場合同一ディレクトリ内に同名のファイルを置くことを許可しているため，ビューアにファイル名を表示するときは read 権も一緒に表示してユーザが区別できるようにする．(U-6)から(U-10)の間はリストファイルをロックして，リストファイルの変更について排他制御する．

4. 実験と考察

本章では，まず初めに計算機実験によって CP-ABE 単体の暗号化と復号の実行時間を測定する．さらに，その結果を用いてファイル名/ディレクトリ名表示やアップロード時の処理の一部について実行時間を考察する．

4.1 CP-ABE の暗号化と復号時間の測定

CP-ABE の暗号化/復号に要する時間を属性数やその結合の方法およびファイルサイズを変更して計測した．CP-ABE のライブラリである cpabe toolkit（cpabe-0.11, libbswabe-0.9）を用いて実験用 PC（OS: Fedora 15, CPU: Pentium 4 3.06GHz, Memory: 1GB）で暗号化/復号を 10 回試行した平均値で実行時間を求めている．処理するファイルは 1byte，1MB，10MB の 3 種類用意し，それぞれに属性を AND のみで結合した場合と OR のみで結合した場合について属性数を変化させて計測した．図 7 は属性数と復号時間の関係，図 8 は属性数と暗号化時間の関係を示している．暗号化時間に関しては属性を AND 結合した場合も OR 結合した場合も同様に属性数に比例して処理時間が増加している．復号時間は OR 結合した場合に属性数が増加しても処理時間が増加していないことが確認できる．また，ハイブリッド型の処理を採用しているため，ファイルサイズを 10MB に増やした場合でも 1 回の暗号化/復号処理の実行時間は 1sec 程度に抑えられることが分かった．

4.2 ファイル名/ディレクトリ名表示時間の考察

ファイルのダウンロード処理の「ファイル名/ディレクトリ名の表示」の時間を考察する．一般ユーザはダウンロード時に 3.3 節の(D-1)～(D-4)を実行してディレクトリの移動およびそのディレクトリ内のファイル名やディレクトリ名を表示させる．この処理は一般ユーザの操作の快適さに影響するため短時間に実行できることが望ましい．

ここで，ある仮定における実行時間を見積もる．リストファイルの各行のデータは

500byte 程度であるとする．さらに，read 権を持つファイルが 1 つのディレクトリに平均で 2000 ファイル程度になると仮定する．この場合，リストファイル 1 つのブロックのサイズは 1MB 程度になる. read 権は復号可能な属性を拡張する OR 結合で増えると仮定すると，1MB の OR の復号時間である 0.2sec 程度で 1 つのブロックが復号可能となる. また, 復号しなければならないリストファイルのブロック数が 10 程度であれば, 0.2×10=2sec でファイル名およびディレクトリ名を表示可能である．また，read 権に AND が多い，例えば 10 個の属性を AND で結合して強い制限を与えている場合でも 1MB のブロックは 0.4sec 程度で復号できる. この場合でも 1 つのブロックを復号するごとにファイル名やディレクトリ名を表示すれば，その影響を小さくできる.

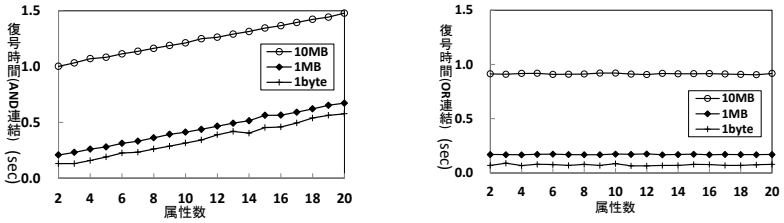

図 7　属性数と復号時間の関係(AND 結合(左), OR 結合(右))

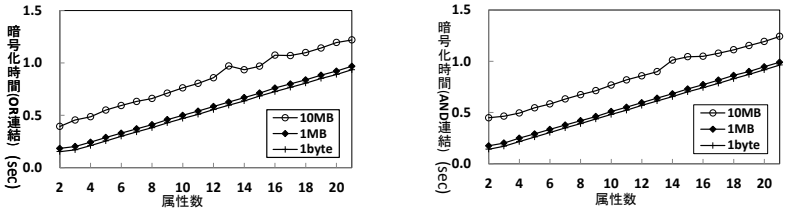

図 8　属性数と暗号化時間の関係(AND 結合(左), OR 結合(右))

4.3 ファイルのアップロード時間の考察

アップロード処理の中でリストファイルがロックする時間を考察する．(U-5)～(U-10)が対応するが，(U-10)のリストファイルのアップロード時間を除けば，(U-6)のリストファイルの復号および(U-9)のリストファイルの暗号化の実行時間が支配的である．(U-10)に関しては同様のシステムで暗号化を課さない場合でも必要な処理であるため，特に CP-ABE により増加する処理時間として（リストファイル復号時間）+（リストファイル暗号化時間）を算出する.

まず，ユーザは(U-1)～(U-2)の操作でアップロードするファイルの read 権および write 権を選択している．したがって，(U-6)のリストファイルの復号は対象となる read 権のブロックを 1 つ選択して実行するだけで良い．4.2 節と同様の仮定で実行時間を見積もる. 1 つのブロックの復号時間は read 権が 10 個の属性を OR で結合された場合には 0.2sec，AND で結合された場合には 0.4sec で復号できる．(U-9)でのリストファイルの暗号化時間も同様に見積もる. 図 8 より暗号化時間は read 権が 10 個の属性を OR で結合する場合と AND で結合する場合で違いはなく，1MB のブロックを暗号化の処理時間はどちらも 0.4sec 程度となる．したがって，提案方式で CP-ABE により増加する処理時間は，本章での仮定の下では 0.6sec～0.8sec 程度と十分小さい値になる.

5. おわりに

本稿では CP-ABE と属性ベース署名を用いてクラウドのストレージ上でファイル名とディレクトリ名の表示制御および書き換え権限を管理する方法を提案した．提案方式はリストファイルによって同一のアクセス権のファイル名/ディレクトリ名の復号を一括して実行することで，それらの表示時間を短縮した．さらに，リストファイルによりディレクトリの論理的な構造とストレージ上のファイルの物理的な構造を関連付けることで複数のアカウントやサーバに分散したファイル群を一つのディレクトリとして扱うことが可能となった．今後の課題はファイル名やディレクトリ名の閲覧/編集権限が変更された場合など詳細な手順の明確化およびシステム全体を通しての実行時間の評価が挙げられる.

謝辞　本研究の一部は科学研究費補助金(23300026)の助成を受けて実施している.

参考文献

[1] Bethencourt, J., Sahai, A. and Waters, B.: Ciphertext-Policy Attribute-Based Encryption, Proc. 2007 IEEE Symposium on Security and Privacy, pp. 321-334 (2007).
[2] Sahai, A. and Waters, B.: Fuzzy Identity-Based Encryption, Proc. EUROCRYPT 2005, LNCS 3493, Springer-Verlag, pp. 457-473 (2005).
[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, 信学技法, LOIS2011-69, vol. 111, no. 383, pp. 73-78 (2012).
[4] Zhao, F., Nishide, T. and Sakurai, K.: Realizing Fine-grained and Flexible Access Control to Outsourced Data with Attribute-based Cryptosystems, Proc. ISPEC 2011, LNCS 6672, Springer-Verlag, pp.83-97 (2011).
[5] Maji, H.K., Prabhakaran, M. and Rosulek, M.: Attribute-Based Signatures, Proc. CT-RSA 2011, LNCS 6558, Springer-Verlag, pp.376-392 (2011).
[6] 後藤めぐ美, 大東俊博, 相原玲二: 属性ベース暗号を利用したファイル名暗号化ビューアの提案, 平成 23 年度電気・情報関連学会中国支部連合大会予稿集, pp.8-9 (2011).