Vol.2019-IOT-44 No.30
2019/3/8

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

石橋 拓哉[1,a)]　小林 海[2]　大東 俊博[2]　土田 光[3]　金岡 晃[4]　柿崎 淑郎[5]　相原 玲二[6]

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．著者らは，複数組織で利用可能なファイル共有システム（MA-ABE）を用いた暗号化システムの提案をしている．しかし，提案システムでの実装・評価は演算処理部分のみ行われているため．本稿では新たに，実現するために必要となる MA-ABE の通信部分を含めた実装・評価を行った．また，実際にこのシステムを用いる場合に生じると考えられる鍵失効に関する問題や，属性管理の問題などについての考察を行った．

キーワード：ファイル共有システム，暗号文ポリシー属性ベース暗号，複数組織対応

Evaluation and Consideration on File Sharing Services using Multi-Authority Attrubute-Based Encryption

Takuya Ishibashi[1,a)]　Kai Kobayashi[2]　Toshihiro Ohigashi[2]　Hikaru Tsuchida[3]
Akira Kanaoka[4]　Yoshio Kakizaki[5]　Reiji Aibara[6]

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．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．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 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 (Key Generation Center: 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

1 東海大学大学院 情報通信学研究科, Graduate School of Information and Telecommunication Engineering，Tokai University
2 東海大学 情報通信学部，School of Information and Telecommunication Engineering，Tokai University
3 日本電気株式会社，NEC
4 東邦大学 理学部，Faculty of Science，Toho University
5 東京電機大学，Tokyo Denki University
6 広島大学 情報メディア教育研究センター，Information Media Center，Hiroshima University
a) t_ishibashi@star.tokai-u.jp
*1 http://www.dropbox.com/

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．KGC は自身が管理している全ユーザの秘密鍵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，複数 KGC に対応できない従来研究では，同一組織内での利用に限定せざるを得なかった．組織外への情報漏えいリスクを低減するには各組織が KGC を個別に運用する要求があることから，複数 KGC への拡張が求められる．

石橋らは 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，複数の KGC が存在可能な属性ベース暗号 (Multi-Authority Attribute-Based Encryption: MA-ABE) を用いた方法について検討している [4]．しかしながら，文献 [4] では暗号化のための公開パラメータのサイズの面で有利な MA-ABE である Yannis らの方式 [5] の処理速度や必要なメモリ量の評価はしているが，KGC やストレージへのアクセス等の通信時間を含めた評価は行っていない．そこで本稿ではユーザと KGC 間の鍵発行にかかる通信部分やデータの暗号化/復号の際に生じるストレージとの通信部分も含めて実装をし，通信時間を含めたシステム全体の評価を行う．さらに，実際に MA-ABE を用いた複数組織間でのシステムの使用を想定した際に問題となると考えられる属性の変化に伴う鍵失効に関する問題について現状の技術に関して調査をし，考察を行う．

2. 準備

本章では各方式の説明で用いる対称ペアリング群について述べた後に，既存のシステムを説明する．まず初めに属性ベース暗号について説明をし，次に従来の CP-ABE についての説明を行う．その後，今回実装を行った複数組織対応属性ベース暗号のシステムを概説する．

2.1 対称ペアリング群

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 p，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．

2.2 属性ベース暗号

2.2.1 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

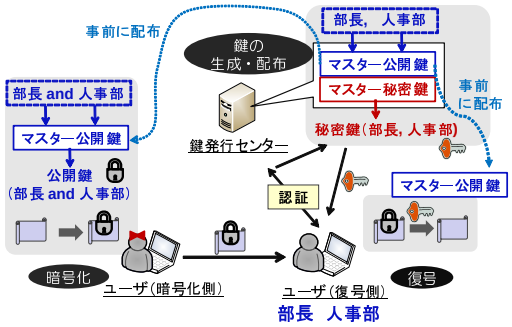

図 1 暗号文ポリシー属性ベース暗号の概要

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とき，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．

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

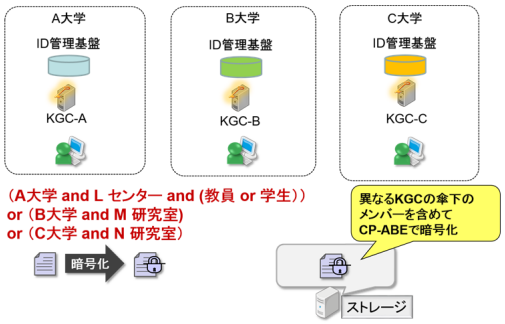

図 2　複数組織での KGC 管理の概要

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．

2.3　複数組織対応属性ベース暗号

クラウドサービスなどの利用をする際には，複数の組織のユーザが共同で利用する場合が考えられる．こういった場合，従来の属性ベース暗号では KGC を複数機関で共同で管理することは非常に難しい．このような場面でも属性ベース暗号を複数機関で利用できるように，KGC を組織ごとに用意して複数機関で連携して使用できる方式である MA-ABE が提案されている（図 2）．MA-ABE には大きく分けて，各機関の KGC を中央機関を使用して管理する方式 [7][8] と中央機関を必要とせず KGC が複数存在可能な方式 [5][9][10][11][12] が存在する．これらの方式は，複数機関で利用する場合などの現実的な属性管理ができるため，従来の鍵発行機関が 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，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．

2.3.1　Yannis らの方式のアルゴリズム

まず初めに Yannis らの方式 [5] のシンタックスを以下に示す．

定義 2.1

Yannis 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．

Yannis 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．このとき，Yannis 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．$\mathrm{GP} = \{\mathsf{param}, H, F, U, U_\Theta, \mathsf{T}\}$ を出力する．

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

KeyGen (GID, θ, u, SK_θ, GP)**:**　鍵発行機関 θ 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．

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

2.4 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

文献 [5] では Yannis らの方式に基づいた共同研究グループでの情報共有システムを検討している．一般的に，情報共有システムは大学単位などの比較的大きな組織ごとに管理がなされている．しかしながら，現実の利用シーン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．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，研究室単位で NDA（秘密保持契約）を締結しているとすると，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，MA-ABE を利用して各研究グループ単位で KGC を管理するようにし，暗号方式で必要となる GID を大学管理の ID 管理基盤の ID を利用する（図 3）．この方式では研究グループ単位での KGC の管理に拡張したことで，システム内に存在する KGC の数は多くなってしまう．しかしながら，Yannis らの方式はグループごとに一つの KGC を用意するだけで使用可能であるため，実用的なパラメータで利用可能となる．

このシステムを複数の大学間のグループで利用できるシステムへ拡張することは，学術認証フェデレーション

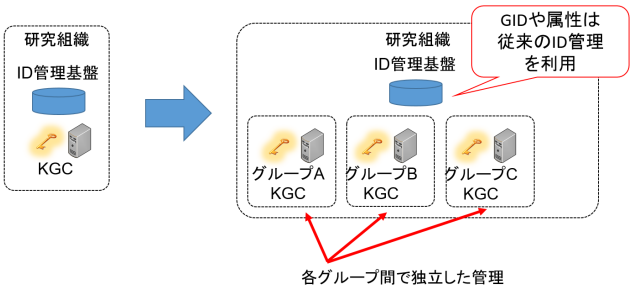

図 3　情報共有システムでの利用

（学認）[*2] を利用して KGC を連携させることにより実現できる．まず学認に参加している自組織の認証プロバイダ（IdP）で認証を行い認可用トークンを発行してもらい，そのトークンを利用してユーザはサービスを利用したい組織の鍵発行センター（KGC=SP）で属性に対応する秘密鍵を発行してもらう．このとき，ユーザ固有の値である GID には ePPN（eduPersonPrincipalName）を利用できる．

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，個々の KGC に任せる方法を取ることにする．KGC を運営している組織について，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ようにする．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，学認が提供している統一した属性を用いても良いし，個々の組織で定義している属性を用いても良いことにする．これらの属性がどの範囲のメンバーを包括するかについては，共同研究等をする関係であれば事前に共有や確認ができることを期待している．しかしながら，属性の範囲の誤解などは起こりうることであり，さらに KGC を設置する組織の大きさ次第では同じ属性を持つメンバーを把握できないことも考えられるため，属性の利用範囲に関しては何らかの工夫が必要になると思われる．

3. 実装

本章では性能評価のために実装したシステムの説明を行う．今回は学術認証フェデレーションとの連携部分は評価の対象とせず，システムの暗号化/復号および，通信部分の評価を行うことを目的として実装を行った．

3.1 暗号化/復号部分の仕様

本システムでは MA-ABE として Yannis らの方式を用いている．Yannis らの方式は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

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

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．

今回の実装では，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．

Authority Setup や KeyGen，Encript 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，鍵発行機関 θ は公開パラメータ $\mathrm{PK} = \{e(g,g)^{\alpha_\theta}, g^{y_\theta}\}$ 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．

3.2 通信部分の仕様

今回使用するシステムではユーザとサーバとの通信によるデータのやりとりを行うための実装も行っている．KGC がユーザからの問い合わせを受けて鍵を発行する処理を行う部分や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が該当する．通信部分の実装では，ユーザ側からはサーバに HTTP/HTTPS リクエストを Python と requests ライブラリを用いて送信するプログラムを実装した．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．これはポート制限の面で HTTP や HTTPS を使い柔軟に管理できるようにしたいためである．今回の図

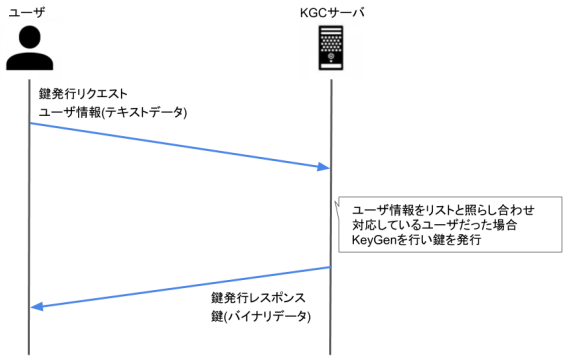

図 4　KGC による鍵発行

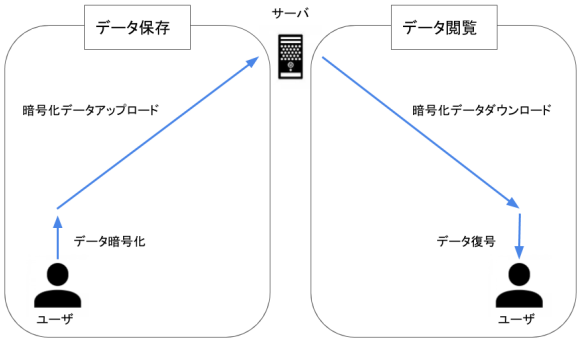

図 5　データの暗号化および復号

表 1　計測に使用したサーバ機器の仕様

CPU	intel(R) Xeon(R) CPU E5-2640 0 @ 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

4, 5 に示したサーバには，さくらのクラウド[*3]の石狩リージョンを利用している．

4. 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 4)．この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 5)．これは，ユーザが手元でデータを暗号化を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）と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．

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

表 2　計測に使用したユーザ機器の仕様

CPU	intel(R) core(TM) 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　鍵発行にかかる通信部分を含めた処理時間 [sec]

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

表 4　暗号化に関する部分の処理時間 [sec]

	暗号化	送信	合計
処理時間	0.4888	0.1176	0.6064

表 5　復号に関する部分の処理時間 [sec]

	受信	復号	合計
処理時間	0.1299	0.2543	0.3842

4.1　KGC の鍵発行に要する処理時間

本節では，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こととする．KGC はこれを受け取り自身が持っている 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，本実験ではユーザの一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 1, 2 の実験環境において 100 回実行した平均処理時間を表 3 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．

4.2　データの保存および閲覧にかかる処理時間

本節では，ユーザがデータを暗号した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てきてから復号するまでの処理時間の計測をそれぞれ行った．

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 4 に示す．この処理時間は，表 1, 2 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 5 に示す．この処理時間は，表 1, 2 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．

5.　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，ユーザの属性変更を考慮する必要がある．ユーザはその時点の属性に基づく秘密鍵を KGC から受け取るが，その後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．以下では対応方法について考察する．

5.1　鍵失効機能を有する ABE を利用する方法

単一の鍵発行機関が存在可能な 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 [13], [14], [15], [16] 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 [17] 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 [18] 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 Key-Policy ABE に対して失効機能を実現した方式 [19] が存在する．しかし，単一の鍵発行機関が存在可能な ABE の利用は複数機関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 [20] 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 [21] 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 [22] 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 [23] も提案されている．

安直には，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 [23] の方式を用い

る方法が良いように思える．しかし，文献 [23] の方式は著者らの知る限り実装が存在しない．また，文献 [23] 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 [23] では，そのような検討は成されていない．

5.2 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〜2019 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．実システムを運用していく上では，属性管理に関して，このように鍵の属性に有効期限を埋め込む方法が有用と思われる．

5.3 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鍵発行機関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 [24] 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 [24] 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，本稿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．

5.4 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 [25] 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．

6. まとめ

まとめ本稿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について通信部分を含めた実装・評価を行った．その結果，実験環境において，鍵発行センターから秘密鍵を取得するまでの処理時間は単一の属性の鍵では 0.31 秒程度であることがわかった．さらに，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，閲覧時には 0.4 秒程度であることがわかった．また，併せて実際にシステムを利用しようとした場合に問題となることが想定される，ユーザの属性管理に関する問題に対する考察および検討をした．その結果，鍵の属性に有効期限を埋め込む方法が有用であることがわかった．

本稿の性能評価により，提案システムはシステムとしての実用性を有しており，今後は運用面へ議論を進めることができるようになったと考えられる．また，運用面に近い議論として鍵失効に関する問題や属性管理問題についての考察を行い，その点でも実用性の明確化に貢献したと言える．

今後の課題は本システムを学術認証フェデレーション上で動作するように実装をし，具体的な利用シーンに基づく評価を行うことである．

謝辞 本研究の一部は JSPS 科研費 JP16H02808 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 Sympo-*

sium 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] 石橋拓哉，大東俊博，　土田光，　金岡晃，柿崎淑郎，相原玲二ほか：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論文集，Vol. 2018, pp. 16–23 (2018).

[5] Rouselakis, Y. and Waters, B.: Efficient Statically-Secure Large-Universe Multi-Authority Attribute-Based Encryption, *Financial Cryptography and Data Security - 19th International Conference, FC 2015, San Juan, Puerto Rico, January 26-30, 2015, Revised Selected Papers*, pp. 315–332 (online), DOI: 10.1007/978-3-662-47854-7_19 (2015).

[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] Müller, S., Katzenbeisser, S. and Eckert, C.: Distributed Attribute-Based Encryption, *Information Security and Cryptology - ICISC 2008, 11th International Conference, Seoul, Korea, December 3-5, 2008, Revised Selected Papers*, pp. 20–36 (online), DOI: 10.1007/978-3-642-00730-9_2 (2008).

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

[10] 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).

[11] 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).

[12] 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).

[13] Sahai, A., Seyalioglu, H. and Waters, B.: Dynamic Credentials and Ciphertext Delegation for Attribute-Based Encryption, *CRYPTO*, Lecture Notes in Computer Science, Vol. 7417, Springer, pp. 199–217 (2012).

[14] Lee, K., Choi, S. G., Lee, D. H., Park, J. H. and Yung, M.: Self-Updatable Encryption: Time Constrained Access Control with Hidden Attributes and Better Efficiency, *ASIACRYPT (1)*, Lecture Notes in Computer Science, Vol. 8269, Springer, pp. 235–254 (2013).

[15] Lee, K., Lee, D. H., Park, J. H. and Yung, M.: CCA Security for Self-Updatable Encryption: Protecting Cloud Data When Clients Read/Write Ciphertexts, (online), DOI: 10.1093/comjnl/bxy122 (2018).

[16] Lee, K., Choi, S. G., Lee, D. H., Park, J. H. and Yung, M.: Self-Updatable Encryption: Time Constrained Access Control with Hidden Attributes and Better Efficiency, *Advances in Cryptology - ASIACRYPT 2013* (Sako, K. and Sarkar, P., eds.), Berlin, Heidelberg, Springer Berlin Heidelberg, pp. 235–254 (2013).

[17] Attrapadung, N. and Imai, H.: Conjunctive Broadcast and Attribute-Based Encryption, *Pairing*, Lecture Notes in Computer Science, Vol. 5671, Springer, pp. 248–265 (2009).

[18] Attrapadung, N. and Imai, H.: Attribute-Based Encryption Supporting Direct/Indirect Revocation Modes, *IMA Int. Conf.*, Lecture Notes in Computer Science, Vol. 5921, Springer, pp. 278–300 (2009).

[19] Datta, P., Dutta, R. and Mukhopadhyay, S.: Adaptively Secure Unrestricted Attribute-Based Encryption with Subset Difference Revocation in Bilinear Groups of Prime Order, *AFRICACRYPT*, Lecture Notes in Computer Science, Vol. 9646, Springer, pp. 325–345 (2016).

[20] Horváth, M.: Attribute-Based Encryption Optimized for Cloud Computing, *SOFSEM*, Lecture Notes in Computer Science, Vol. 8939, Springer, pp. 566–577 (2015).

[21] Tsuchida, H., Nishide, T., Okamoto, E. and Kim, K.: Revocable Decentralized Multi-Authority Functional Encryption, *INDOCRYPT*, Lecture Notes in Computer Science, Vol. 10095, pp. 248–265 (2016).

[22] NOMURA, K., MOHRI, M., SHIRAISHI, Y. and MORII, M.: Attribute Revocable 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 (online), DOI: 10.1587/transinf.2016OFP0004 (2017).

[23] Li, D., Chen, J., Liu, J., Wu, Q. and Liu, W.: Efficient CCA2 Secure Revocable Multi-authority Large-Universe Attribute-Based Encryption, *Cyberspace Safety and Security* (Wen, S., Wu, W. and Castiglione, A., eds.), Cham, Springer International Publishing, pp. 103–118 (2017).

[24] Watanabe, D., Sakazaki, H. and Miyazaki, K.: Representative System and Security Message Transmission using Re-encryption Scheme Based on Symmetric-key Cryptography, *Journal of Information Processing*, Vol. 25, pp. 67–74 (online), DOI: 10.2197/ipsjjip.25.67 (2017).

[25] 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).