2Z-3

更新用の属性鍵とプロキシ再暗号化により属性失効処理を分散させる属性ベース暗号

成瀬　猛†　　毛利　公美‡　　白石　善明†　　野口　亮司††

†名古屋工業大学　　‡岐阜大学　　††（株）豊通シスコム

1. はじめに

クラウドストレージを利用したデータ共有では，クラウドからの情報漏えいを防ぐためにデータを暗号化することが望まれる．

暗号文ポリシー属性ベース暗号（Ciphertext-Policy ABE：CP-ABE）では，暗号文に復号条件を表す属性に関するアクセス構造を埋め込むことで，復号者をグループで指定することができる．アクセス構造を満たす属性が含まれている秘密鍵を持つユーザは暗号文を復号することができる．

属性ベース暗号では一般に，属性を失効したユーザが暗号文を復号し続けられることを防ぐために，データ所有者は新しい公開鍵で共有データを再暗号化しなければならない．また，システム権威者は再暗号化された共有データを他のユーザが復号できるように新しい秘密鍵を送信しなければならない．単純な属性失効では，再暗号化をする場合にはデータ所有者が，秘密鍵の再配布をする場合にはシステム権威者が処理を行うことになる．

プロキシ再暗号化（Proxy Re-Encryption：PRE）や二重暗号化を利用して，属性失効処理をクラウドサーバおよびクラウドサービスプロバイダに分散する方式 [1][2][3]が提案されている．文献[1][2][3]の方式では，秘密鍵に属性を追加できず，属性の変更はできない．属性の追加や変更をしたければ新しい秘密鍵の発行が必要である．

本稿では，PRE を利用して属性失効処理をクラウドサーバに分散し，ユーザの属性を追加・変更するときに新しい秘密鍵の発行が不要な CP-ABE を提案する．CP-ABE は Waters の方式[4]をベースとする．Waters の方式は dqPBDHE 仮定のもとで Selective-set モデルにおいて IND-CPA 安全であることが証明されている．

2. システムに求められる要件

文献[1]の方式では，ユーザのある属性を失効させる場合，すべての暗号文に対するアクセス権を失わせて，システム権威者はユーザが失効しない属性を含む新しい秘密鍵を送信することになる．このことから，多人数のユーザが利用できるシステムにするためには次の課題が挙げられる．

【課題 1】システム権威者が容易にユーザの属性を変更できる

【課題 2】ユーザの属性を追加するときのユーザの鍵発行申請とシステム権威者の秘密鍵の発行・送信処理を無くす

これらの課題に対応する要件は次のようになる．

【要件 1】ユーザの属性を指定して失効できる

【要件 2】ユーザの秘密鍵に属性を追加できる

3. 提案方式

3.1. エンティティ

提案方式は“ユーザ”，“データ所有者”，“システム権威者”，“クラウドサーバ”で構成される．

【ユーザ】クラウドサーバに保存されている暗号化された共有データをダウンロードし，アクセス構造を満たす属性を持つユーザは復号できる．

【データ所有者】共有データを暗号化し，クラウドサーバにアップロードする．

【システム権威者】暗号化に必要な鍵を公開し，ユーザの属性が含まれている秘密鍵と，再暗号化・秘密鍵の更新に用いる PRE 鍵を発行する．プロトコルに従い，いかなる不正も行わない機関である．

【クラウドサーバ】共有データを保管しており，システム権威者から受け取った PRE 鍵を用いて再暗号化・秘密鍵の更新をする．クラウドサーバは curious-but-honest[1][2][3]であり，プロトコルに従うが，共有データの情報を知ろうとする． honest であるため，ユーザとの結託攻撃はしない．

3.2. アルゴリズム

Auth.Setup：

システム内の属性の数Uを入力とする

1. 素数 q，素数位数 q の 2 つの群G_1, G_2，生成元$P \in G_1$，双線形写像$e: G_1 \times G_1 \to G_2$を選ぶ
2. U個の属性と関連づいた群の要素$h_1, \dots, h_U \in G_1$，属性鍵を示す乱数$t_1, \dots, t_U, u_1, \dots, u_U \in Z_q$を選ぶ
3. 乱数$a, \alpha \in Z_q$を選ぶ
4. 公開鍵$PK := (P, e(P,P)^{\alpha}, aP, h_1, \dots, h_U, T_1 = t_1 P, \dots, T_U)$，マスター秘密鍵$MSK := (\alpha P, t_1, \dots, t_U, u_1, \dots, u_U)$，PRE 鍵$rk_{0\to 1} := \{rk_{i(0)\to i(1)}\}_{1\le i\le U} = \{\frac{t_1}{u_1}, \dots, \frac{t_U}{u_U}\}$を出力する．

Auth.Ext：

マスター秘密鍵MSK，ユーザの属性集合Sを入力する

1. 乱数$t \in Z_q$を選ぶ
2. $SK := (S, K, L, \forall_{x \in U} K_x) = \left(S, \alpha P + atP, tP, x \in S\ K_x = \frac{t}{t_x} h_x, x \notin S\ K_x = \frac{t}{u_x} h_x\right)$を秘密鍵として出力する

CKEnc：

メッセージ$\mathcal{M}$，コンテンツ鍵Kを入力し，暗号文$Enc_K(\mathcal{M})$を出力する

DO.Enc：

公開鍵PK，LSSS アクセス構造(M, ρ)，コンテンツ鍵Kを入力する

1. Mを$l \times n$行列とする
   ベクトル$\vec{v} := (s, y_2, \dots, y_n) \in Z_q^n$をランダムに選ぶ
2. $\lambda_i := \vec{v} \cdot M_i (i = 1, \dots, l)$を求める
3. 乱数$r_1, \dots, r_l \in Z_q$を選ぶ
4. $C_K := (CS, C, C', (C_1, D_1), \dots, (C_l, D_l)) = (\{\rho(1), \dots, \rho(l)\}, Ke(P,P)^{\alpha s}, sP, (\lambda_1(aP) - r_1 h_{\rho(1)}, r_1 T_{\rho(1)}), \dots, (\lambda_l(aP) - r_l h_{\rho(l)}, r_l T_{\rho(l)}))$を暗号化されたコンテンツ鍵として出力する

U.Dec：

ユーザの秘密鍵SK，暗号化されたコンテンツ鍵C_Kを入力する

1. $I \subset \{1,2, \dots, l\}, I = \{i : \rho(i) \in S\}$と定義し，$\{\lambda_i\}$を$s$の正しいシェアとして$\sum_{i \in I} \omega_i \lambda_i = s$となる$\{\omega_i \in Z_q\}_{i \in I}$を求める
2. $\frac{e(C', K)}{\prod_{i \in I} \left(e(C_i, L) e(D_i, K_{\rho(i)})\right)^{\omega_i}} = \frac{e(P,P)^{\alpha s} e(P,P)^{ast}}{\prod_{i \in I} (e(P,P)^{ta\lambda_i \omega_i})} = e(P,P)^{\alpha s}$を計算する
3. $\frac{Ke(P,P)^{\alpha s}}{e(P,P)^{\alpha s}} = K$を計算し，$K$をコンテンツ鍵として出力する

CKDec：

暗号文$Enc_K(\mathcal{M})$，コンテンツ鍵Kを入力し，メッセージ$\mathcal{M}$を出力する

Auth.ReKeyGen：

ユーザが失効した属性の集合γ，マスター秘密鍵MSKを入力する

1. 各属性鍵の現在のバージョンをn_xとし，バージョンが

Attribute-based Encryption Sharing Attribute Revocation Process with Service Server by Proxy Re-Encryption and Attribute Key for Updating

† Takeru NARUSE and Yoshiaki SHIRAISHI・Nagoya Institute of Technology

‡ Masami MOHRI・Gifu University

†† Ryoji NOGUCHI・Toyotsu Syscom Corp.

n_xである属性xの属性鍵を$t_{x(n_x)}$とする
新しい属性鍵 $\forall_x \in \theta\ t_{x(n_x+1)} \in Z_p$をランダムに選ぶ
2. $\forall_x \in \theta\ T'_x \coloneqq t_{x(n_x+1)}P$を計算する
3. 再定義されたマスター秘密鍵MSK'，再定義された公開鍵PK'，PRE鍵$\forall_x \in \theta\ rk_{x(n_x)\to x(n_x+1)} \coloneqq \frac{t_{x(n+1)}}{t_{x(n)}}$を出力する

C.ReEnc：
更新する属性$y = \rho(i)$，暗号化されたC_Kの成分D_i，属性yのPRE鍵リストRKL_yを入力する
1. 現在の属性鍵のバージョンを確認し，最新のバージョンである場合は⊥を出力しアルゴリズムを終了する
2. 現在の属性鍵のバージョンを $n-i$，最新の属性鍵のバージョンを n とする
$rk_{y(n-i)\to y(n)} \coloneqq rk_{y(n-i)\to y(n-i+1)} \cdot \ldots \cdot rk_{y(n-1)\to y(n)} = \frac{t_{y(n)}}{t_{y(n-i)}}$を計算する
3. $D'_i \coloneqq rk_{y(n-i)\to y(n)} \cdot D_i = \frac{t_{y(n)}}{t_{y(n-i)}} \cdot r_i t_{y(n-i)} P = r_i t_{y(n)} P$ を計算し，再暗号化された暗号文C'_Kの成分として出力する

C.ReKey：
更新する属性w，秘密鍵SKの成分K_w，属性wのPRE鍵リストRKL_wを入力する
1. 現在の属性鍵のバージョンを確認し，最新のバージョンである場合は⊥を出力しアルゴリズムを終了する
2. 現在の属性鍵のバージョンを $n-i$，最新の属性鍵のバージョンを n とする．
$rk_{w(n-i)\to w(n)} \coloneqq rk_{w(n-i)\to w(n-i+1)} \cdot \ldots \cdot rk_{w(n-1)\to w(n)} = \frac{t_{w(n)}}{t_{w(n-i)}}$を計算する．
3. $K'_w \coloneqq rk^{-1}_{w(n-i)\to w(n)} \cdot K_w = \frac{t_{w(n-i)}}{t_{w(n)}} \cdot \frac{t}{t_{w(n-i)}} h_w = \frac{t}{t_{w(n)}} h_w$を計算し，更新された秘密鍵$SK$の成分として出力する

提案方式の流れを図 1 に示す．

3.3. 提案方式における Waters の方式[4]からの変更点

【変更点 1】更新用の属性鍵を生成し，暗号文と秘密鍵に含める
クラウドサーバに再暗号化と秘密鍵の更新処理を分散させるためのものが更新用の属性鍵である．PRE 鍵を用いてクラウドサーバが属性鍵を更新することで，再暗号化（C.ReEnc）・秘密鍵の更新（C.ReKey）をする．

また，提案方式では文献[1]の方式のように，クラウドサーバに秘密鍵の成分を保存するためにユーザの秘密鍵にはダミー属性を埋め込んでいる．すべての暗号文のアクセス構造は，アクセス木で表現すると，ルートノードが AND ゲートで，かつルートノードの子ノードの 1 つはダミー属性（Att_D）が割り当てられている葉ノードになっている．クラウドサーバは属性鍵t_xを更新するために，クラウドサーバは属性に関連づいた各ユーザの秘密鍵のうちのいくつかの成分K_xを保存しているが，ダミー属性に対応した秘密鍵の成分K_{AttD}を知らないため，暗号文を復号することはできない．

【変更点 2】システム内の各属性について，ポジティブ(ユーザが与えられている)な属性とネガティブ(ユーザが与えられていない)な属性に対応する属性鍵を生成して秘密鍵に含める
クラウドサーバが秘密鍵に属性を追加できるようにするためのものである．

システム権威者はネガティブに対応する属性鍵（バージョン 0）をポジティブに対応する属性鍵（バージョン 1）に変換する PRE 鍵$rk_{0\to1}$をクラウドサーバに送信する．クラウドサーバが PRE 鍵$rk_{0\to1}$を用いて属性鍵を変換することで秘密鍵に属性を追加する．

4. 安全性

アクセス構造を満たす属性を持っていないユーザとクラウドサーバが，暗号化されたコンテンツ鍵C_Kからコンテンツ鍵Kを得ることができないことを示す．

ポジティブな属性とネガティブな属性では対応している属性鍵が異なるため，アクセス構造を満たす属性を持っていないユーザは暗号化されたコンテンツ鍵C_Kを復号することはできない．また，Waters の方式[4]をベースとしている提案方式は結託耐性を持つ．アクセス構造を満たす属性を持っていないユーザ同士が結託しても秘密鍵を組み合わせることはできないため，暗号化されたコンテンツ鍵C_Kを復号することはできない．

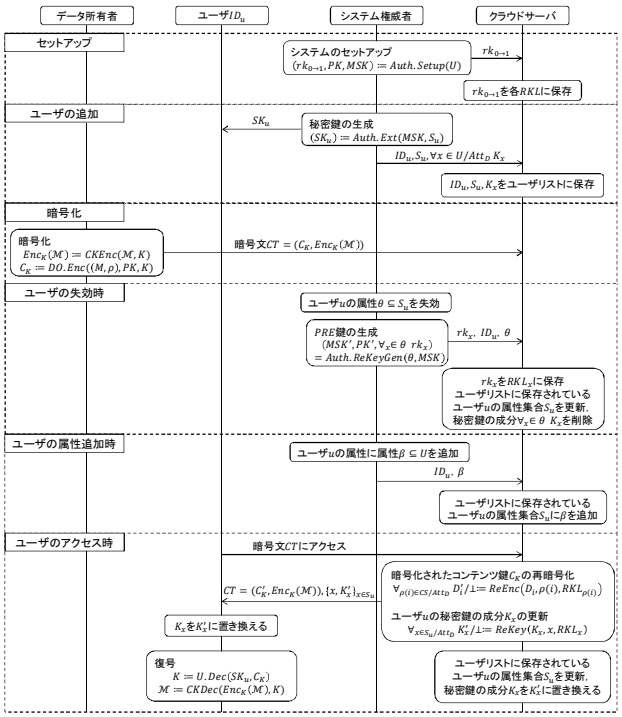

図 1　提案方式の流れ

表 1　関連研究との比較

	提案方式	S.Yuら[1]	S.Yuら[2]	J.Huら[3]
暗号方式	CP-ABE	KP-ABE	CP-ABE	CP-ABE
要件1	○	×	○	○
要件2	○	×	×	×

クラウドサーバは，ダミー属性以外の属性に対応する各ユーザの秘密鍵の成分K_xを保存している． 暗号化されたコンテンツ鍵C_Kを復号するにはダミー属性に対応する秘密鍵の成分K_{AttD}が必要であるため，復号することができない．

5. 比較とまとめ

関連研究との比較を表 1 に示す．

要件 1 に対しては，提案方式ではユーザが失効した属性以外の属性鍵をクラウドサーバが更新することでユーザの属性を指定して失効できる．

要件 2 に対しては，提案方式では秘密鍵に含まれるネガティブな属性に対応する属性鍵をポジティブな属性に対応する属性鍵にクラウドサーバが変換することで，ユーザの属性を追加することができる．

以上のように，本稿では，PRE を利用して属性失効処理をクラウドサーバに分散し，ユーザの属性を追加・変更するときに新しい秘密鍵の発行が不要な CP-ABE を提案した．

厳密な安全性証明をすることが今後の課題である．

参考文献

[1] S. Yu, C. Wang, K. Ren, and W. Lou: Achieving Secure, Scalable, and Fine-grained Data Access Control in Cloud Computing, In Proceedings of the 29th IEEE International Conference on Computer Communications, pp.534-542 (2010).
[2] S. Yu, C. Wang, K. Ren, and W. Lou: Attribute Based Data Sharing with Attribute Revocation, In Proceedings of the 5th ACM Symposium on Information, Computer and Communications Security, pp.261-270 (2010).
[3] J. Hur and D. K. Noh: Attribute-Based Access Control with Efficient Revocation in Data Outsourcing Systems, IEEE Transactions on Parallel and Distributed Systems, Vol.22, No.7, pp.1214-1221 (2011).
[4] B. Waters: Ciphertext-Policy Attribute-Based Encryption: An Expressive, Efficient, and Provably Secure Realization, In Proceedings of the 14th International Conference on Practice and Theory in Public Key Cryptography, pp.53-70 (2011).