

推薦論文

# 前方秘匿性を満たす属性失効機能付き属性ベース暗号

成瀬 猛<sup>1</sup> 毛利 公美<sup>2</sup> 白石 善明<sup>3,a)</sup>

受付日 2013年11月29日, 採録日 2014年7月11日

**概要:** クラウドストレージに保存した共有データのアクセス制御に適した暗号として, 暗号文ポリシ属性ベース暗号 (Ciphertext-Policy Attribute-Based Encryption: CP-ABE) がある. CP-ABE では, 秘密鍵に関連付けられている属性集合が, 暗号文に関連付けられているアクセス構造を満たす場合のみ, その秘密鍵によって暗号文を復号することができる. 属性ベース暗号ではユーザの属性を失効させるには, そのユーザが暗号文を復号できないようにしなければならない. 効率良くユーザの属性を失効させることができる CP-ABE の提案がなされているが, 属性失効ユーザに対する厳密な安全性証明が与えられていない, もしくは, ジェネリック群モデルという強い仮定のモデルのもとで証明されている. 本稿では, 前方秘匿性 (Forward Secrecy) を満たす属性失効機能付き属性ベース暗号を提案する. 属性ベース暗号における前方秘匿性は, 一度属性を失効したユーザはそこから先は暗号文を復号できないことを意味する. 提案方式は不正ユーザ, クラウドサーバ, 属性失効ユーザの攻撃に対して, 標準モデルのもと DBDH 仮定において IND-CPA 安全である.

**キーワード:** 暗号化クラウドストレージ, 暗号文ポリシ属性ベース暗号, プロキシ再暗号化, 属性失効

## Attribute Revocable Attribute-Based Encryption with Forward Secrecy

TAKERU NARUSE<sup>1</sup> MASAMI MOHRI<sup>2</sup> YOSHIAKI SHIRAIISHI<sup>3,a)</sup>

Received: November 29, 2013, Accepted: July 11, 2014

**Abstract:** Ciphertext-Policy Attribute-Based Encryption (CP-ABE) is suitable for access control of shared data in a cloud storage. In CP-ABE, a user is able to decrypt ciphertext with his/her secret key as long as the set of attributes embedded in the secret key satisfies the access structure associated with the ciphertext. In ABE, to revoke attributes of a user, it is necessary to make unable to decrypt ciphertext. Some CP-ABE schemes for efficient attribute revocation have been proposed. However, they have not been given formal security proof against a revoked user or have been proved in the generic group model that is a strong assumption model. This paper proposes an attribute-revocable CP-ABE with forward secrecy. Forward secrecy in ABE means that once attributes belonging to a user is revoked, the user cannot decrypt the ciphertext. The proposed scheme is IND-CPA secure against attack by unauthorized user, cloud servers and revoked user in the DBDH assumption under the standard model.

**Keywords:** cryptographic cloud storage, ciphertext-policy attribute-based encryption, proxy re-encryption, attribute revocation

### 1. はじめに

システムの運用費用などのコストの削減や, 利便性の点からクラウドコンピューティングが注目されている. クラ

本稿の内容は 2013 年 7 月のマルチメディア, 分散, 協調とモバイル (DICOMO2013) シンポジウム 2013 にて報告され, コンピュータセキュリティ研究会主査により情報処理学会論文誌ジャーナルへの掲載が推薦された論文である.

<sup>1</sup> 名古屋工業大学  
Nagoya Institute of Technology, Nagoya, Aichi 466-8555, Japan

<sup>2</sup> 岐阜大学  
Gifu University, Gifu 501-1193, Japan

<sup>3</sup> 神戸大学  
Kobe University, Kobe, Hyogo 657-8501, Japan

a) zenmei@port.kobe-u.ac.jp

クラウドサービスの1つに、クラウドストレージを利用したデータ共有サービスがあるが、サービス提供側のサーバにデータを保管するため、外部からの不正アクセスやサービス提供者による不正行為などの情報漏えいが懸念される。情報漏えいを防ぐために、共有データを暗号化しクラウドストレージに保存する Cryptographic Cloud Storage (CCS) であることが望ましい。共有データのアクセス制御に適した暗号として属性ベース暗号 (Attribute-Based Encryption: ABE) がある。

ABEは公開鍵暗号の一種であり、2005年に Sahaiら [1]によって提案された。ABEのうち、暗号文ポリシー属性ベース暗号 (Ciphertext-Policy Attribute Based Encryption: CP-ABE) [2], [3]では、暗号文ごとに復号者を属性に基づいたグループで指定することができる。暗号文に属性に関する復号条件を表すアクセス構造が定められ、ユーザが持つ秘密鍵にそのユーザが有する属性の集合が関連付けられている。秘密鍵に関連付けられている属性集合が暗号文に定められているアクセス構造を満たす場合のみ、その秘密鍵によって暗号文を復号することができる。暗号化するときアクセス構造を定めることで、暗号文ごとに復号者を属性に基づいたグループで指定することができる CP-ABEは、共有データの細かいアクセス制御に適している。

ABEでは、ユーザの属性を失効させるには、そのユーザが暗号文を復号できないようにしなければならない。単純な方法として、パラメータを更新した公開鍵ですべての共有データの暗号化をやり直す方法があげられる。しかし、あるユーザが属性を失効するたびにすべての共有データを暗号化し直すのは現実的ではない。文献 [4], [5], [6]では効率良くユーザの属性を失効させることができる ABEの提案がなされている。

Yuらの方式 [4]は、属性を失効したユーザが暗号文を復号できないようにするための再暗号化をプロキシサーバに任せるプロキシ再暗号化方式である。再暗号化はユーザが暗号化された共有データを要求したときに行う。更新前のパラメータで生成された秘密鍵では再暗号化文を復号できない。再暗号化文を他のユーザが復号できるように、プロキシサーバがユーザの秘密鍵を更新する。秘密鍵の更新は、パラメータの更新後にユーザが初めて共有データを要求したときに行う。この方式では、属性を失効したユーザが再暗号化された暗号文を復号できないことの厳密な証明はされていない。

Hurらの方式 [5]では、クラウドサービス提供側であるデータマネージャが、属性ベース暗号の公開鍵で暗号化された共有データをランダムに選んだ属性グループ鍵を使ってさらに暗号化することで属性を失効したユーザが暗号文を復号できないようにしている。各ユーザにはそれぞれ異なる共通鍵である KEK (Key Encryption Key) が配られ

ている。データマネージャは、属性を失効していないユーザが属性グループ鍵でさらに暗号された暗号化共有データを復号できるように、KEKを使って属性グループ鍵を暗号化し、ユーザに送信する。ユーザの数だけ KEKがあるため、ユーザが多数の状況では鍵管理が煩雑になる。この方式も厳密な安全性証明はされていない。

Ostrovskyらは文献 [6]で、Pirrettiらが提案したブロードキャスト失効方式 [7]を Bethencourtが提案した CP-ABE [3]に適用できることを示している。この方式における属性の失効では、ユーザの属性を指定して失効することはできない。ユーザが持つ複数の属性のうち1つだけを失効させたい場合でも、すべての属性を失効させなければならない。また、文献 [3]の方式の安全性は、標準モデルより強い仮定であるジェネリック群モデルのもとで証明されている。

本稿では、ユーザの属性を指定して失効することができる、前方秘匿性を持つ属性失効機能付き属性ベース暗号を提案する。提案方式の安全性は標準モデルのもとで証明する。属性ベース暗号における前方秘匿性は、一度属性を失効したユーザはそこから先は暗号文を復号できないことを意味する [5]。また、提案方式では前方秘匿性に加えて、文献 [2], [3], [4], [5], [6]を参考に次の2つの性質を持つことをセキュリティ要件とする。

1. データの機密性：アクセス構造を満たす属性を有していない不正なユーザとクラウドサーバは、共有データの平文にアクセスすることはできない。
2. 結託耐性：複数人の不正なユーザとクラウドサーバが結託しても、アクセス構造を満たしていない暗号文を復号することはできない。

攻撃モデル1を不正ユーザとクラウドサーバが結託するモデル、攻撃モデル2を属性失効ユーザの攻撃モデルとする。各攻撃モデルにおいて、提案方式が標準モデルのもと Decisional Bilinear Diffie-Hellman (DBDH) 仮定において IND-CPA 安全であることを証明し、セキュリティ要件を満たすことを示す。

## 2. 準備

### 2.1 双線形写像

$G_1, G_2$ を素数位数  $p$ の有限巡回群とする。 $G_1$ の生成元  $P \in G_1$ 、任意の  $a, b \in \mathbb{Z}_p$ に対し、双線形写像  $e: G_1 \times G_1 \rightarrow G_2$ は以下の2つの性質を満たす。

1. 双線形性： $e(aP, bP) = e(P, P)^{ab}$
2. 非退化性： $e(P, P) \neq 1$

$G_1$ 上の群演算と上記の双線形写像  $e: G_1 \times G_1 \rightarrow G_2$ が効率的に計算可能であれば、 $G_1$ は双線形群であるという。また、 $(aP, bP) = e(P, P)^{ab} = e(bP, aP)$ より、双線形写像は対称性を持つ。

## 2.2 Decisional Bilinear Diffie-Hellman (DBDH) 仮定

$a, b, c, z \in \mathbb{Z}_p$  をランダムに選び,  $P \in G$  を生成元とする. DBDH 仮定とは,  $(A = aP, B = bP, C = cP, Z = e(P, P)^{abc})$  と  $(A = aP, B = bP, C = cP, Z = e(P, P)^z)$  を無視できないアドバンテージで識別することができる多項式時間アルゴリズム  $\mathcal{A}$  は存在しないという仮定である. このとき,  $\mathcal{A}$  が  $Z = e(P, P)^{abc}$  であると推測したときに 0 を出力するときの DBDH 問題に対するアドバンテージを

$$\text{Adv}(\mathcal{A}) = |\Pr[\mathcal{A}(P, A, B, C, e(P, P)^{abc}) = 0] - \Pr[\mathcal{A}(P, A, B, C, e(P, P)^z) = 0]|$$

とする.

## 2.3 暗号文ポリシー属性ベース暗号 (CP-ABE)

CP-ABE は 2007 年に文献 [3] で Bethencourt らによって提案された. CP-ABE では暗号文に復号条件を表すアクセス構造が, 秘密鍵にユーザの属性集合が関連付けられている. 一般的な CP-ABE は, Setup, Encrypt, Extract, Decrypt の 4 つのアルゴリズムからなる.

- Setup: セキュリティパラメータ  $\lambda$  を入力とし, 公開鍵  $PK$  とマスター秘密鍵  $MK$  を出力する.
- Extract: マスター秘密鍵  $MK$  とユーザの属性集合  $S$  を入力とし, 秘密鍵  $SK$  を出力する.
- Encrypt: 公開鍵  $PK$  と平文  $M$ , アクセス構造  $W$  を入力とし, 暗号文  $CT$  を出力する.
- Decrypt: 暗号文  $CT$ , 秘密鍵  $SK$  を入力とし,  $SK$  を生成する際に使った属性集合  $S$  がアクセス構造  $W$  を満たす場合, 平文  $M$  を出力する.

## 2.4 CP-ABE の安全性

CP-ABE の Selective-structure モデルにおける IND-CPA 安全性は, 以下の攻撃者  $\mathcal{A}$  と挑戦者  $\mathcal{C}$  とのゲームによって定義される [2].

- Init. 攻撃者  $\mathcal{A}$  はチャレンジアクセス構造  $W^*$  を挑戦者  $\mathcal{C}$  に送る.
- Setup. 挑戦者  $\mathcal{C}$  はセットアップアルゴリズムを実行し, 公開鍵  $PK$  を攻撃者  $\mathcal{A}$  に送る.
- Phase1. 攻撃者  $\mathcal{A}$  は以下のクエリを行うことができる.
  - Ext クエリ: 攻撃者  $\mathcal{A}$  は,  $W^*$  を満たさない属性集合  $S$  を挑戦者  $\mathcal{C}$  に送り, 挑戦者  $\mathcal{C}$  は  $S$  に対応する秘密鍵  $SK$  を攻撃者  $\mathcal{A}$  に送る.
- Challenge. 攻撃者  $\mathcal{A}$  は, 同じ長さの平文  $M_0, M_1$  を挑戦者  $\mathcal{C}$  にクエリする. 挑戦者  $\mathcal{C}$  は, ランダムに  $\mu \in \{0, 1\}$  を選び,  $W^*$  を暗号化鍵として  $M_\mu$  を暗号化し, 暗号文  $CT^*$  を攻撃者  $\mathcal{A}$  に送る.
- Phase2. 攻撃者は, Phase1 と同様のクエリを行うことができる.

- Guess. 攻撃者は  $\mu$  の推測値  $\mu' \in \{0, 1\}$  を出力する. このゲームにおける攻撃者  $\mathcal{A}$  のアドバンテージを

$$\text{Adv}(\mathcal{A}) = \left| \Pr[\mu' = \mu] - \frac{1}{2} \right|$$

とする. 任意の確率的多項式時間アルゴリズム  $\mathcal{A}$  に対して  $\text{Adv}(\mathcal{A})$  が無視できるほど小さいとき, この暗号は IND-CPA 安全であるという.

## 3. システムのモデル

### 3.1 エンティティ

提案方式は “ユーザ”, “データ所有者”, “属性管理機関”, “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管されている共有データをダウンロードする.

[データ所有者]

データを暗号化し, クラウドサーバにアップロードする.

[属性管理機関]

システム内の属性を管理し, 暗号化に必要な鍵を公開している. ユーザの属性が含まれている秘密鍵と, 再暗号化に使う再暗号化鍵を発行する. プロトコルに従い, いかなる不正も行わない信頼できる機関である.

[クラウドサーバ]

共有データを保管している. 再暗号化鍵を使ってその共有データを再暗号化する. クラウドサーバは curious-but-honest [4], [5] であり, プロトコルに従うが, 暗号化された共有データの情報を可能な限り知ろうとする.

### 3.2 システムの構成とアルゴリズムの定義

提案方式は, Auth.Setup, Auth.Ext, DO.Enc, C.ReEnc, U.Dec の 5 つのアルゴリズムからなる. Auth.Setup と Auth.Ext は属性管理機関, DO.Enc はデータ所有者, C.ReEnc はクラウドサーバ, U.Dec はユーザによって実行される.

- Auth.Setup: セキュリティパラメータ  $\lambda$  を入力として, 公開鍵  $PK$  とマスター秘密鍵  $MK$ , 再暗号化鍵  $RK$  を出力する.
  - Auth.Ext: マスター秘密鍵  $MK$  とユーザの属性集合  $S$  を入力として, 秘密鍵  $SK$  を出力する.
  - DO.Enc: 公開鍵  $PK$  と平文  $M$ , アクセス構造  $W$  を入力として, 暗号文  $CT'$  を出力する.
  - C.ReEnc: 暗号文  $CT'$  とユーザの属性集合  $S$ , 再暗号化鍵  $RK$  を入力として, 再暗号化文  $CT$  を出力する.
  - U.Dec: 秘密鍵  $SK$  と再暗号化文  $CT$  を入力として,  $SK$  を生成する際に使った属性集合  $S$  がアクセス構造  $W$  を満たすのであれば, 平文  $M$  を出力する.
- 図 1 に提案方式のモデルを示す.

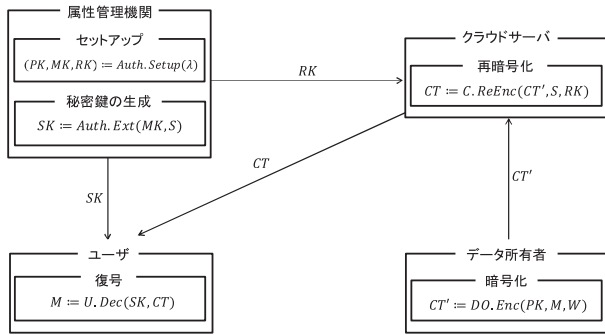


図 1 提案方式のモデル

Fig. 1 Model of the proposed scheme.

### 3.3 アクセス構造

全体の属性数を  $n$  とし、属性空間を  $U = \{1, 2, \dots, n\}$  とする。提案方式のアクセス構造は、1つの AND ゲートのみで構成されるものとし、アクセス構造  $W$  を  $W = \bigwedge_{i \in I} \dot{i}$  と表す。 $\dot{i}$  は属性  $i$  のリテラル ( $i$  または  $\neg i$ ) である。 $\dot{i} = i$  は、復号時に属性  $i$  を持っていなければならないこと (positive) を示す。 $\dot{i} = \neg i$  は、復号時に属性  $i$  を持っていないこと (negative) を示す。属性集合  $I$  は、アクセス構造  $W$  において positive または negative である属性  $i$  の集合である。 $i \notin I$  である場合は、復号時に属性  $i$  について考慮しない (以下、don't care と表す)。また、 $S \models W$  は属性集合  $S$  がアクセス構造  $W$  を満たすことをあらわし、 $S \not\models W$  は満たさないことをあらわす。

### 3.4 安全性の定義

暗号化された共有データの機密性が保たれていることと提案方式が結託耐性と前方秘匿性を持つことを示す。データの機密性を崩す攻撃者は、不正ユーザとクラウドサーバである。攻撃モデル 1 を不正ユーザとクラウドサーバが結託する攻撃モデル、攻撃モデル 2 を属性失効ユーザによる攻撃モデルとする。クラウドサーバは暗号化された共有データの情報をできる限り知ろうとするが、honest であるため属性失効ユーザとは結託しないとする [4], [5]。各攻撃モデルにおいて IND-CPA 安全であることを証明することで、データの機密性が保たれていることと前方秘匿性を持つことを示す。

#### 3.4.1 攻撃モデル 1 の定義

不正ユーザとクラウドサーバが結託するモデルである。攻撃モデル 1 における IND-CPA 安全を以下の攻撃者 A と挑戦者 C とのゲームによって定義する。

- Init. 攻撃者 A はチャレンジアクセス構造  $W^*$  を挑戦者 C に送る。
- Setup. 挑戦者 C は Auth.Setup アルゴリズムを実行し、攻撃者 A に公開パラメータ  $PK$ , 再暗号化鍵  $RK$  を送る。
- Phase1. 攻撃者 A は以下のクエリを行うことができる。
  - Ext クエリ: 攻撃者 A は、 $W^*$  を満たさない属性集

合  $S$  を挑戦者 C に送り、挑戦者 C は  $S$  に対応する秘密鍵  $SK$  を攻撃者 A に送る。

- Challenge. 攻撃者 A は、同じ長さの平文  $M_0, M_1$  を挑戦者 C にクエリする。始めに、挑戦者 C はランダムに  $\mu \in \{0, 1\}$  を選び、 $W^*$  を暗号化鍵として  $M_\mu$  を暗号化し、暗号文  $CT'^*$  を生成する。次に、挑戦者 C は C.ReEnc アルゴリズムを実行し、再暗号化文  $CT^*$  を生成し、攻撃者 A に送る。

- Phase2. 攻撃者 A は、Phase1 と同様のクエリを行うことができる。

- Guess. 攻撃者 A は  $b$  の推測値  $\mu' \in \{0, 1\}$  を出力し、 $\mu' = \mu$  ならば攻撃者 A の勝ちとする。

このゲームにおける攻撃者 A のアドバンテージを

$$\text{Adv}(A) = \left| \Pr[\mu' = \mu] - \frac{1}{2} \right|$$

とする。任意の確率的多項式時間アルゴリズム A に対して  $\text{Adv}(A)$  が無視できるほど小さいとき、この暗号は攻撃モデル 1 において IND-CPA 安全であるという。

#### 3.4.2 攻撃モデル 2 の定義

属性失効ユーザが攻撃を行うモデルである。攻撃モデル 2 における IND-CPA 安全を以下の攻撃者 A と挑戦者 C とのゲームによって定義する。

- Init. 攻撃者 A はチャレンジアクセス構造  $W^*$  と失効属性  $x^*$  を挑戦者 C に送る。ただし、失効属性  $x^*$  はチャレンジアクセス構造  $W^*$  において  $x^* \in I$  を満たす。
- Setup. 挑戦者 C は Setup アルゴリズムを実行し、攻撃者 A に公開鍵  $PK$  を送る。
- Phase1. 攻撃者 A は以下のクエリを行うことができる。

- Ext クエリ: 攻撃者 A は、属性集合  $S$  を挑戦者 C に送り、挑戦者 C は  $S$  に対応する秘密鍵  $SK$  を攻撃者 A に送る。ただし、属性集合  $S$  は、チャレンジアクセス構造  $W^*$  において失効属性  $x^*$  が  $\underline{x^*} = x^*$  である場合は  $x^* \in S$  を、 $\underline{x^*} = \neg x^*$  である場合は  $x^* \notin S$  を満たす。
- ReEnc クエリ: 攻撃者 A は暗号文  $CT'$  と属性集合  $S_R$  を挑戦者 C に送り、挑戦者 C は再暗号化文  $CT^*$  を返す。ただし、属性集合  $S_R$  は、チャレンジアクセス構造  $W^*$  において失効属性  $x^*$  が  $\underline{x^*} = x^*$  である場合は  $x^* \notin S_R$  を、 $\underline{x^*} = \neg x^*$  である場合は  $x^* \in S_R$  を満たす。

- Challenge. 攻撃者 A は、同じ長さの平文  $M_0, M_1$  を挑戦者 C にクエリする。始めに、挑戦者 C はランダムに  $\mu \in \{0, 1\}$  を選び、 $W^*$  を暗号化鍵として  $M_\mu$  を暗号化し、暗号文  $CT'^*$  を生成する。次に、挑戦者 C は ReEnc アルゴリズムを実行し、再暗号化文  $CT^*$  を生成し、攻撃者 A に送る。このとき、ReEnc アルゴリズムに入力する属性集合は、次の条件を満たす。チャレンジアクセス構造  $W^*$  において、 $\dot{i} = i$  である属性  $i$  の集合を  $S^*$  と

定義する. ReEnc アルゴリズムに入力する属性集合は,  $x^* \in S^*$  の場合は  $S^* - \{x^*\}$ ,  $x^* \notin S^*$  の場合は  $S^* \cup \{x^*\}$  である.

-Phase2. 攻撃者 A は, Phase1 と同様のクエリを行うことができる.

-Guess. 攻撃者 A は  $\mu$  の推測値  $\mu' \in \{0, 1\}$  を出力し,  $\mu' = \mu$  ならば攻撃者 A の勝ちとする.

このゲームにおける攻撃者 A のアドバンテージを

$$\text{Adv}(A) = \left| \Pr[\mu' = \mu] - \frac{1}{2} \right|$$

とする. 任意の確率的多項式時間アルゴリズム A に対して  $\text{Adv}(A)$  が無視できるほど小さいとき, この暗号は攻撃モデル 2 において IND-CPA 安全であるという.

## 4. 提案方式

本章では, 前方秘匿性を持つ属性失効機能付き属性ベース暗号を提案する. なお, モデルの定義は 3.2 節に従うものとする.

### 4.1 提案方式の設計方針

提案方式は, Cheung らが提案した方式 (CN07) [2] をベースに構成されている. CN07 は, DBDH 仮定において IND-CPA 安全であることが証明されている.

提案方式において, 属性管理機関が生成した公開鍵で暗号化された暗号文は, そのままではユーザの秘密鍵で復号することはできない. ユーザが共有暗号化データをクラウドサーバからダウンロードするときに, クラウドサーバがプロキシ再暗号化を利用して暗号文を再暗号化することで, ユーザが復号できるようにする. もしも属性を失効したユーザがダウンロードしたときに, 暗号文の一部を送信しないことで復号されることを防ぐようにした場合, クラウドサーバが不完全な暗号文を送ったのか, 正しい暗号文が送信途中で壊れたのかユーザには区別がつかないため, 提案方式では暗号文の一部を再暗号化しないことで復号されることを防ぐ.

### 4.2 アルゴリズム

提案方式のアルゴリズムの詳細を示す.

-Auth.Setup: セキュリティパラメータ  $\lambda$  を入力とする. システム全体の属性空間を  $\mathcal{U} = \{1, \dots, n\}$  とする. まず, 素数  $p$ , 素数位数  $p$  の群  $G_1, G_2$ , 生成元  $P \in G_1$ , 双線形写像  $e: G_1 \times G_1 \rightarrow G_2$  を選ぶ. 次に,  $y, t_1, \dots, t_{3n}, d_1, \dots, d_{2n} \in \mathbb{Z}_p$  をランダムに選び, 以下を計算する.

$$Y := e(P, P)^y.$$

$$T_i := \begin{cases} d_i P & (1 \leq i \leq 2n). \\ t_i P & (2n+1 \leq i \leq 3n). \end{cases}$$

$$rk_i := \frac{t_i}{d_i} \quad (1 \leq i \leq 2n).$$

公開鍵  $PK := (e, P, Y = e(P, P)^y, T_1, \dots, T_{3n})$  とマスター秘密鍵  $MK := (y, d_1, \dots, d_{2n}, t_1, \dots, t_{3n})$ , 再暗号化鍵  $RK := (rk_1, \dots, rk_{2n})$  を出力する.

-Auth.Ext: マスター秘密鍵  $MK$  とユーザの属性集合  $S$  を入力とする. まず, 各  $i \in \mathcal{U}$  について,  $r_i \in \mathbb{Z}_p$  をランダムに選び,  $r := \sum_{i=1}^n r_i$  とする. 次に,  $\hat{D} := (y - r)P$  を計算し, 各  $i \in \mathcal{U}$  について  $D_i, F_i$  を以下のように計算する.

$$D_i := \begin{cases} \frac{r_i}{t_i} P & (i \in S). \\ \frac{r_i}{t_{n+i}} P & (i \notin S). \end{cases}$$

$$F_i := \frac{r_i}{t_{2n+i}} P.$$

秘密鍵  $SK := (\hat{D}, \{(D_i, F_i) | i \in \mathcal{U}\})$  を出力する.

-DO.Enc: 公開鍵  $PK$  と平文  $M$ , アクセス構造  $W$  を入力とする. まず, ランダムに  $s \in \mathbb{Z}_p$  を選ぶ. 次に,  $\tilde{C} = M \cdot Y^s$ ,  $\hat{C} := sP$  を計算し, 各  $i \in \mathcal{U}$  について  $C_i$  を以下のように計算する.

$$C'_i := \begin{cases} sT_i & (i \in I \wedge \underline{i} = i). \\ sT_{n+i} & (i \in I \wedge \underline{i} = \neg i). \\ sT_{2n+i} & (i \notin I). \end{cases}$$

暗号文  $CT' := (W, \tilde{C}, \hat{C}, \{C'_i | i \in \mathcal{U}\})$  を出力する.

-C.ReEnc: 暗号文  $CT'$ , ユーザの属性集合  $S$ , 再暗号化鍵  $RK$  を入力とする. 各  $i \in \mathcal{U}$  について以下のように計算する.

$i \in S \wedge (i \in I \wedge \underline{i} = i)$  の場合,

$$C_i := rk_i \cdot C'_i = \frac{t_i}{d_i} \cdot s \cdot d_i P = s \cdot t_i P.$$

$i \notin S \wedge (i \in I \wedge \underline{i} = \neg i)$  の場合,

$$C_i := rk_{n+i} \cdot C'_i = \frac{t_{n+i}}{d_{n+i}} \cdot s \cdot d_{n+i} P = s \cdot t_{n+i} P.$$

それ以外の属性については,  $C_i := C'_i$  とする.

再暗号化文  $CT := (W, \tilde{C}, \hat{C}, \{C_i\})$  を出力する.

-U.Dec: 秘密鍵  $SK$ , 再暗号化文  $CT$  を入力する. 各  $i \in \mathcal{U}$  について以下のように計算する.

$i \in S \wedge \underline{i} = i$  の場合,

$$e(C_i, D_i) := e\left(s \cdot t_i P, \frac{r_i}{t_i} P\right) = e(P, P)^{r_i \cdot s}.$$

$i \notin S \wedge \underline{i} = \neg i$  の場合,

$$e(C_i, D_i) := e\left(s \cdot t_{n+i} P, \frac{r_i}{t_{n+i}} P\right) = e(P, P)^{r_i \cdot s}.$$

$i \notin I$  の場合,

$$e(C_i, F_i) := e\left(s \cdot t_{2n+i} P, \frac{r_i}{t_{2n+i}} P\right) = e(P, P)^{r_i \cdot s}.$$

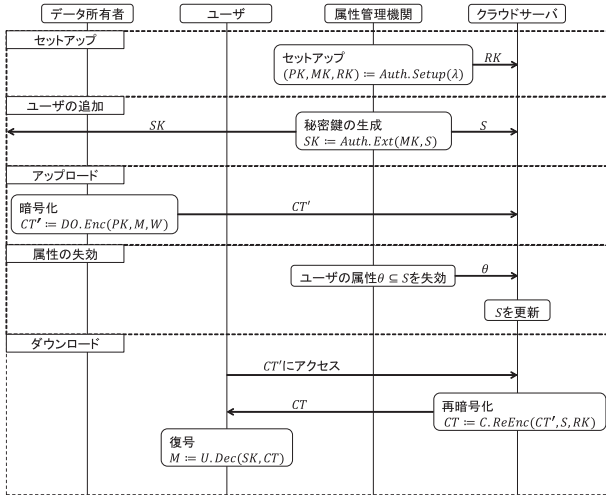


図 2 提案方式の流れ  
Fig. 2 Flow of the proposed scheme.

次に、以下のように平文  $M$  を復号する。

$$e(P, P)^{y \cdot s} := e(sP, (y - r)P) \cdot e(P, P)^{r \cdot s}$$

$$= e(\hat{C}, \hat{D}) \cdot \prod_{i=1}^n e(P, P)^{r_i \cdot s}$$

$$\frac{\hat{C}}{e(P, P)^{y \cdot s}} = \frac{\hat{C}}{Y^s} = M.$$

平文  $M$  を出力する。

以上のアルゴリズムを用いた提案方式の流れを図 2 に示す。属性失効の手順を次節で示す。

### 4.3 属性の失効手順

属性の失効は、次の 2 つのフェーズによって行われる。なお、属性の失効に必要な再暗号化鍵と各ユーザの属性集合  $S$  は、セッアップとユーザの追加を行う際に、属性管理機関がクラウドサーバにあらかじめ送信されている。

[フェーズ 1]

属性管理機関は、ユーザの属性を失効する際に、失効したユーザの属性の情報をクラウドサーバに送信し、クラウドサーバは受け取った情報をもとに、保持しているユーザの属性集合  $S$  を更新する。

[フェーズ 2]

クラウドサーバは、属性を失効したユーザがクラウドサーバ上の暗号文にアクセスした際に、暗号文と再暗号化鍵  $RK$ 、Step3 で更新したユーザの属性集合  $S$  を入力として  $ReEnc$  アルゴリズムを実行し、ユーザが再暗号化文を復号できないように暗号文の一部を除いて再暗号化し、ユーザに再暗号化文を送る。

## 5. 安全性証明

3.4 節の安全性定義に従って、各攻撃モデルにおける安全性の証明をする。

表 1 公開鍵要素  $T_i$  の計算

Table 1 Computation of the public components  $T_i$ .

	$i \in I$		$i \notin I$ (don't care)
	$i = i$ (positive)	$i = \neg i$ (negative)	
$T_i$ (positive)	$\alpha_i P$	$\alpha_i B$	$\alpha_i B$
$T_{n+i}$ (negative)	$\beta_i B$	$\beta_i P$	$\beta_i B$
$T_{2n+i}$ (don't care)	$\gamma_i B$	$\gamma_i B$	$\gamma_i P$

### 5.1 攻撃モデル 1 の証明

不正ユーザとクラウドサーバが結託するモデルである。攻撃モデル 1 において、提案方式が IND-CPA 安全であることを証明する。

[定理 1] DBDH 問題が困難ならば、提案方式は攻撃モデル 1 において IND-CPA 安全である。

[証明] 攻撃者  $A$  は、提案方式の攻撃モデル 1 に対して、無視できないアドバンテージ  $\epsilon$  を持つと仮定する。このとき、少なくとも  $\epsilon/2$  のアドバンテージで DBDH 問題を解くシミュレータが構成可能であることを示す。DBDH 問題を出題する挑戦者  $C$  に対し、DBDH 問題を解くシミュレータ  $B$  を作成する。

まず、挑戦者  $C$  は素数位数  $p$  の群  $G_1, G_2$  を選び、双線形写像  $e$  と生成元  $P \in G_1$  を決める。次に、 $a, b, c, z \in Z_p$  と  $v \in \{0, 1\}$  を選ぶ。  $v = 0$  ならば  $(A, B, C, Z) = (aP, bP, cP, e(P, P)^{abc})$ 、  $v = 1$  ならば  $(A, B, C, Z) = (aP, bP, cP, e(P, P)^z)$  として、  $(A, B, C, Z)$  をシミュレータ  $B$  に送る。

-Init. 攻撃者  $A$  はシミュレータ  $B$  にチャレンジアクセス構造  $W^* = \bigwedge_{i \in I} \hat{i}$  を送る。

-Setup. シミュレータ  $B$  は公開鍵  $PK$  を生成する。  $Y := e(A, B) = e(P, P)^{ab}$  とする。すべての  $i \in U$  について、  $\alpha_i, \beta_i, \gamma_i \in Z_p$  をランダムに選び、表 1 に従って、  $T_i, T_{n+i}, T_{2n+i}$  を設定し、再暗号化鍵  $rk_i \in Z_p$  ( $1 \leq i \leq 2n$ ) をランダムに選ぶ。シミュレータ  $B$  は攻撃者  $A$  に公開鍵  $PK$  と再暗号化鍵  $RK$  を送る。

-Phase1. 攻撃者  $A$  は以下のクエリを行うことができる。

-Ext クエリ：攻撃者  $A$  は、属性集合  $S \subseteq U$  をシミュレータ  $B$  にクエリする。ただし、属性集合  $S$  は  $S \neq W^*$  を満たすものとする。このとき、  $j \in S \wedge \hat{j} = \neg j$  または  $j \notin S \wedge \hat{j} = j$  のいずれかを満たす属性  $j \in I$  が必ず存在する。今回の証明では、一般性より後者の条件を満たす属性  $j$  を選ぶ。まず、シミュレータ  $B$  は、すべての  $i \in U$  について、  $r'_i \in Z_p$  をランダムに選ぶ。  $r_j := ab + r'_j \cdot b$  とし、すべての  $i \neq j$  について、  $r_i := r'_i \cdot b$  とする。そして、  $r := \sum_{i=1}^n r_i = ab + \sum_{i=1}^n r'_i \cdot b$  とする。次に、以下のように  $\hat{D}, D_i$  を計算する。

$$\hat{D} := \prod_{i=1}^n 1/(r'_i \cdot B) = \left( - \sum_{i=1}^n r'_i \cdot b \right) P$$

$$= (ab - r)P.$$

$i = j$  について,

$$D_j := \frac{1}{\beta_j} A \cdot \frac{r'_j}{\beta_j} P = \frac{ab + r'_j \cdot b}{b \cdot \beta_j} P = \frac{r_j}{b \cdot \beta_j} P.$$

$i \neq j$  について,  $i \in S$  である場合,

$$(1) \quad i \in I \wedge \underline{i} = i. \quad D_i := \frac{r'_i}{\alpha_i \cdot rk_i} B = \frac{r_i}{\alpha_i \cdot rk_i} P.$$

$$(2) \quad (i \in I \wedge \underline{i} = \neg i) \vee i \notin I. \quad D_i := \frac{r'_i}{\alpha_i \cdot rk_i} P = \frac{r_i}{b \cdot \alpha_i \cdot rk_i} P.$$

$i \notin S$  である場合,

$$(1) \quad (i \in I \wedge \underline{i} = i) \vee i \notin I. \quad D_i := \frac{r'_i}{\beta_i \cdot rk_{n+i}} P = \frac{r_i}{b \cdot \beta_i \cdot rk_{n+i}} P.$$

$$(2) \quad i \in I \wedge \underline{i} = \neg i. \quad D_i := \frac{r'_i}{\beta_i} B = \frac{r_i}{\beta_i} P.$$

次のように  $F_i$  を計算する.

$i = j$  の場合,

$$F_j := \frac{1}{\gamma_i} A \cdot \frac{r'_j}{\gamma_j} P = \frac{ab + r'_j \cdot b}{b \cdot \gamma_j} P = \frac{r_j}{b \cdot \gamma_j} P.$$

$i \neq j$  の場合

$$(1) \quad i \in I. \quad F_i := \frac{r'_i}{\gamma_i} P = \frac{r_i}{b \cdot \gamma_i} P.$$

$$(2) \quad i \notin I. \quad F_i := \frac{r'_i}{\gamma_i} B = \frac{r_i}{\gamma_i} P.$$

シミュレータ B は秘密鍵  $SK$  を攻撃者 A に送る.

-Challenge. 攻撃者 A はシミュレータ B に, 同じ長さの平文  $M_0, M_1$  を送る. シミュレータは  $\mu \in \{0, 1\}$  をランダムに選ぶ.  $\tilde{C} := M_\mu \cdot Z$  とする. シミュレータ B は次のように再暗号化文  $CT^*$  を計算し, 攻撃者 A に送る.

$$CT^* := (W, \tilde{C}, C, \{rk_i \alpha_i C | i \in I \wedge \underline{i} = i\},$$

$$\{rk_i \beta_i C | i \in I \wedge \underline{i} = \neg i\}, \{\gamma_i C | i \notin I\}).$$

-Phase2. 攻撃者 A は Phase1 と同じクエリを実行することができる.

-Guess. 攻撃者 A は  $\mu$  を推測し, 推測値  $\mu' \in \{0, 1\}$  をシミュレータ B に送る. シミュレータ B は, 攻撃者 A の推測が正しければ, DBDH 問題の組が正しいもの ( $Z = e(P, P)^{abc}$ ) だとして  $v' = 0$  を, 正しくなければ  $Z = e(P, P)^z$  だとして  $v' = 1$  を出力する.

$Z = e(P, P)^{abc}$  である場合, 暗号文  $CT^*$  は正しい暗号文であり, 攻撃者 A のアドバンテージは  $\epsilon$  である. よって,

$$\Pr[v' = 0 | Z = e(P, P)^{abc}] = \Pr[\mu' = \mu | Z = e(P, P)^{abc}] = \frac{1}{2} + \epsilon.$$

$Z = e(P, P)^z$  の場合,  $\tilde{C}$  は攻撃者 A から見て完全にランダムなものとなる. 攻撃者 A のアドバンテージは 0 であり, 推測はランダムとなる. よって,

$$\Pr[v' = 1 | Z = e(P, P)^z] = \Pr[\mu' \neq \mu | Z = e(P, P)^z] = \frac{1}{2}$$

以上より, シミュレータ B は DBDH 問題に対して少なくとも  $\epsilon/2$  以上のアドバンテージを持つ.

## 5.2 攻撃モデル 2 の証明

属性失効ユーザが攻撃を行うモデルである. 攻撃モデル 2 において, 提案方式が IND-CPA 安全であることを証明する.

[定理 2] DBDH 問題が困難ならば, 提案方式は攻撃モデル 2 において IND-CPA 安全である.

[証明] 攻撃者 A は, 提案方式の攻撃モデル 2 に対して, 無視できないアドバンテージ  $\epsilon$  を持つと仮定する. このとき, 少なくとも  $\epsilon/2$  のアドバンテージで DBDH 問題を解くシミュレータが構成可能であることを示す. DBDH 問題を出題する挑戦者 C に対し, DBDH 問題を解くシミュレータ B を作成する.

まず, 挑戦者 C は素数位数  $p$  の群  $G_1, G_2$  を選び, 双線形写像  $e$  と生成元  $P \in G_1$  を決める. 次に,  $a, b, c, z \in Z_p$  と  $v \in \{0, 1\}$  を選ぶ.  $v = 0$  ならば  $(A, B, C, Z) = (aP, bP, cP, e(P, P)^{abc})$ ,  $v = 1$  ならば  $(A, B, C, Z) = (aP, bP, cP, e(P, P)^z)$  として,  $(A, B, C, Z)$  をシミュレータ B に送る.

-Init. 攻撃者 A はシミュレータ B にチャレンジアクセス構造  $W^* = \bigwedge_{i \in I} \underline{i}$  と失効属性  $x^* \in \mathcal{U}$  を送る. ただし, 属性  $x^*$  は,  $x^* \in I$  を満たすものとする.

-Setup. シミュレータ B は公開鍵  $PK$  を生成する.  $Y := e(A, B) = e(P, P)^{ab}$  とする. すべての  $i \in \mathcal{U}$  について,  $\alpha_i, \beta_i, \gamma_i \in Z_p$  をランダムに選び, 表 1 に従って  $T_i, T_{n+i}, T_{2n+i}$  を設定する. 属性  $x^*$  について,  $z_{x^*} \in Z_p$  をランダムに選び,  $x^*$  が  $x^* \in I \wedge \underline{x^*} = x^*$  である場合は  $rk_{x^*} := (z_{x^*} \cdot b) / \alpha_{x^*}$ ,  $x^* \in I \wedge \underline{x^*} = \neg x^*$  である場合は  $rk_{x^*} := (z_{x^*} \cdot b) / \beta_{x^*}$  とする. すべての  $i \neq x^*$  について, 再暗号化鍵  $rk_i \in Z_p$  をランダムに選ぶ. シミュレータ B は攻撃者 A に公開鍵  $PK$  を送る.

-Phase1. 攻撃者 A は以下のクエリを行うことができる.

-Ext クエリ: 攻撃者 A は, 属性集合  $S \subseteq \mathcal{U}$  をシミュレータ B にクエリする. ただし, 属性集合  $S$  は  $x^*$  が  $x^* \in I \wedge \underline{x^*} = x^*$  である場合は  $x^* \in S$  を,  $x^* \in I \wedge \underline{x^*} = \neg x^*$  である場合は  $x^* \notin S$  を満たすものとする. まず, シミュレータ B は, すべての  $i \in \mathcal{U}$  について,  $r'_i \in Z_p$  をランダムに選ぶ.  $r_{x^*} := ab + r'_{x^*} \cdot b$  とし, すべての  $i \neq x^*$  について,  $r_i := r'_i \cdot b$  とする. そして,  $r := \sum_{i=1}^n r_i = ab + \sum_{i=1}^n r'_i \cdot b$  とする. 次に, 以下のように  $\hat{D}, D_i$  を計算する.

$$\hat{D} := \prod_{i=1}^n 1/(r'_i \cdot B) = \left( - \sum_{i=1}^n r'_i \cdot b \right) P = (ab - r)P.$$

$i = x^*$  について,  $x^* \in I \wedge \underline{x^*} = x^*$  である場合,

$$D_{x^*} := \frac{1}{z_{x^*}} A \cdot \frac{r'_{x^*}}{z_{x^*}} P = \frac{ab + r'_{x^*} \cdot b}{b \cdot z_{x^*}} P = \frac{r_{x^*}}{\alpha_{x^*} \cdot rk_{x^*}} P.$$

$x^* \in I \wedge \underline{x^*} = \neg x^*$  である場合,

$$D_{x^*} := \frac{1}{z_{x^*}} B \cdot \frac{r'_{x^*}}{z_{x^*}} P = \frac{ab + r'_{x^*} \cdot b}{a \cdot z_{x^*}} P = \frac{r_{x^*}}{\beta_{x^*} \cdot rk_{x^*}} P.$$

$i \neq x^*$  について,  $i \in S$  である場合,

$$(1) i \in I \wedge \underline{i} = i. D_i := \frac{r'_i}{\alpha_i \cdot rk_i} B = \frac{r_i}{\alpha_i \cdot rk_i} P.$$

$$(2) (i \in I \wedge \underline{i} = \neg i) \vee i \notin I. D_i := \frac{r'_i}{\alpha_i \cdot rk_i} P = \frac{r_i}{b \cdot \alpha_i \cdot rk_i} P.$$

$i \notin S$  である場合,

$$(1) (i \in I \wedge \underline{i} = i) \vee i \notin I. D_i := \frac{r'_i}{\beta_i \cdot rk_{n+i}} P = \frac{r_i}{b \cdot \beta_i \cdot rk_{n+i}} P.$$

$$(2) i \in I \wedge \underline{i} = \neg i. D_i := \frac{r'_i}{\beta_i \cdot rk_{n+i}} B = \frac{r_i}{\beta_i \cdot rk_{n+i}} P.$$

次のように  $F_i$  を計算する.

$i = x^*$  の場合,

$$F_{x^*} := \frac{1}{\gamma_{x^*}} A \cdot \frac{r'_{x^*}}{\gamma_{x^*}} P = \frac{ab + r'_{x^*} \cdot b}{b \cdot \gamma_{x^*}} P = \frac{r_{x^*}}{b \cdot \gamma_{x^*}} P.$$

$i \neq x^*$  の場合

$$(1) i \in I. F_i := \frac{r'_i}{\gamma_i} P = \frac{r_i}{b \cdot \gamma_i} P.$$

$$(2) i \notin I. F_i := \frac{r'_i}{\gamma_i} B = \frac{r_i}{\gamma_i} P.$$

-ReEnc クエリ: 攻撃者 A は暗号文  $CT'$  と属性集合  $S_R$  をシミュレータ B にクエリする. ただし, 属性集合  $S_R$  は  $x^* \in I \wedge \underline{x^*} = x^*$  である場合は  $x^* \notin S_R$  を,  $x^* \in I \wedge \underline{x^*} = \neg x^*$  である場合は  $x^* \in S_R$  を満たすものとする. シミュレータ B は各  $i \in S_R \wedge (i \in I \wedge \underline{i} = i)$ , または各  $i \notin S_R \wedge (i \in I \wedge \underline{i} = \neg i)$  について,  $C_i := rk_i \cdot C'_i$  を計算し, 再暗号化文  $CT$  を攻撃者 A に送る.

-Challenge. 攻撃者 A はシミュレータ B に, 同じ長さの平文  $M_0, M_1$  を送る. シミュレータは  $\mu \in \{0, 1\}$  をランダムに選ぶ.  $\tilde{C} := M_\mu \cdot Z$  とする. シミュレータ B は次のように再暗号化文  $CT^*$  を以下のように計算し, 攻撃者 A に送る.

$x^* \in I \wedge \underline{x^*} = x^*$  である場合,

$$CT^* := (W, \tilde{C}, C, \{rk_i \alpha_i C | i \neq x^* \wedge (i \in I \wedge \underline{i} = i)\}, \{\alpha_i C | i = x^* \wedge (i \in I \wedge \underline{i} = i)\}, \{rk_i \beta_i C | i \in I \wedge \underline{i} = \neg i\}, \{\gamma_i C | i \notin I\}).$$

$x^* \in I \wedge \underline{x^*} = \neg x^*$  である場合,

$$CT^* := (W, \tilde{C}, C, \{rk_i \alpha_i C | i \neq x^* \wedge (i \in I \wedge \underline{i} = i)\}, \{\beta_i C | i = x^* \wedge (i \in I \wedge \underline{i} = \neg i)\}, \{rk_i \beta_i C | i \in I \wedge \underline{i} = \neg i\}, \{\gamma_i C | i \notin I\}).$$

-Phase2. Phase1 と同様のクエリを行うことができる.

-Guess. 攻撃者 A は  $\mu$  を推測し, 推測値  $\mu' \in \{0, 1\}$  をシミュレータ B に送る. シミュレータ B は, 攻撃者 A の推測が正しければ, DBDH 問題の組が正しいもの ( $Z = e(P, P)^{abc}$ ) だとして  $v' = 0$  を, 正しくなければ  $Z = e(P, P)^z$  だとして  $v' = 1$  を出力する.

$Z = e(P, P)^{abc}$  である場合, 暗号文  $CT^*$  は正しい暗号文であり, 攻撃者 A のアドバンテージは  $\epsilon$  である.

よって,

$$\Pr[v' = 0 | Z = e(P, P)^{abc}] = \Pr[\mu' = \mu | Z = e(P, P)^{abc}] = \frac{1}{2} + \epsilon.$$

$Z = (P, P)^z$  の場合,  $\tilde{C}$  は攻撃者 A から見て完全にランダムなものとなる. 攻撃者 A のアドバンテージは 0 であり, 推測はランダムとなる. よって,

$$\Pr[v' = 1 | Z = e(P, P)^z] = \Pr[\mu' \neq \mu | Z = e(P, P)^z] = \frac{1}{2}$$

以上より, シミュレータ B は DBDH 問題に対して少なくとも  $\epsilon/2$  以上のアドバンテージを持つ.

## 6. 性能評価

属性を指定した失効が可能な Yu らの方式 [4], Hur らの方式 [5] と提案方式を鍵とメッセージのサイズ, 計算量, 安全性について比較を行った結果を表 2 に示す. Ostrovsky らの方式 [6] は属性を指定した失効ができないためにここでは比較の対象としていない.

公開鍵, 秘密鍵, 暗号文のサイズは Hur らの方式が最も小さく, Yu らの方式と提案方式は同じである. 再暗号化鍵のサイズは属性の失効回数が 1 回以上, ユーザ数がシステム全体の属性数より多いときに提案方式が最も小さくなる.

暗号化と秘密鍵生成の計算量は, Hur らの方式が少なく, Yu らの方式と提案方式は同じである. 再暗号化の計算量は, 更新する属性の数が同じ場合は三方式が等しくなる. 秘密鍵の更新の計算量は, Yu らの方式と Hur の方式は等しく, 提案方式はゼロである. ここで, 再暗号化で更新する属性の数が等しいという状況で秘密鍵の更新を考える. Yu らの方式と Hur らの方式の秘密鍵の更新はユーザがアクセスしたときに行われる. ユーザがアクセスした際にクラウドサーバが行う再暗号化と秘密鍵の更新を

表 2 従来方式との比較

Table 2 Comparison of schemes.

	提案方式	Yu らの方式[4]	Hur らの方式[5]
公開鍵のサイズ	$(3 U  + 1) \times  G  +  G_r $	$(3 U  + 1) \times  G  +  G_r $	$2 \times  G  +  G_r $
秘密鍵のサイズ	$(2 U  + 1) \times  G $	$(2 U  + 1) \times  G $	$(2 S  + 1) \times  G  + \log N  \times  K $
暗号文のサイズ	$( U  + 1) \times  G  +  G_r $	$( U  + 1) \times  G  +  G_r $	$(2 U  + 1) \times  G  +  G_r $
再暗号化鍵のサイズ	$2 U  \times  Z_p $	$2r U  \times  Z_p $	$(2 N  - 1) \times  K $
暗号化の計算量	$( U  + 2) \times \exp$	$( U  + 2) \times \exp$	$(2 U  + 2) \times \exp$
秘密鍵生成の計算量	$(2 U  + 1) \times \exp$	$(2 U  + 1) \times \exp$	$(2 S  + 2) \times \exp$
再暗号化の計算量	$ R  \times \exp$	$ R_{CT}  \times \exp$	$ R_{CT}  \times \exp$
秘密鍵更新の計算量	なし	$ R_{SK}  \times \exp$	$ R_{SK}  \times \exp$
復号の計算量	$( U  + 1) \times \hat{e} + ( U  + 1) \times \exp$	$( U  + 1) \times \hat{e} + ( U  + 1) \times \exp$	$(2 U  + 1) \times \hat{e} + (2 U  + 2) \times \exp$
攻撃モデル1 (3.3.1) に対する証明可能安全性 (結託耐性)	IND-CPA安全 (DBDH仮定)	IND-CPA安全 (DBDH仮定)	—
攻撃モデル2 (3.3.2) に対する証明可能安全性 (前方秘匿性)	IND-CPA安全 (DBDH仮定)	—	—

exp: G 上のべき乗演算,  $\hat{e}$ : ペアリング演算, |U|: システム全体の属性数, |S|: ユーザが有する属性数, |R|: 復号条件が一致している属性の数, r: 属性失効イベントが起きた回数, |R<sub>SK</sub>|: 更新する属性の数(秘密鍵), |R<sub>CT</sub>|: 更新する属性の数(暗号文), |N|: ユーザの総数, |U|: アクセス構造に含まれる属性の数, |K|: 共通鍵のサイズ



合計した計算量は、Yuらの方式とHurらの方式は等しく、それらよりも提案方式の方が小さくなる。つまり、属性の一部の失効が頻繁に起こるときには、クラウドサーバでの計算の負荷は提案方式が最も低い。

安全性について、提案方式は不正ユーザと属性失効ユーザの両方に厳密な証明がされている。

## 7. おわりに

本稿では、前方秘匿性を持つ属性失効機能付き属性ベース暗号を提案した。提案方式は、ユーザの属性を指定して失効することができる。提案方式は、各攻撃モデルにおいて、標準モデルのもとDBDH仮定においてIND-CPA安全であることを証明することで、データの機密性、結託耐性、前方秘匿性を満たすことを示した。

提案方式で扱えるアクセス構造はANDゲート1つのみの構成であるため、より自由度の高いアクセス構造を扱うようにすることが今後の課題としてあげられる。

謝辞 本研究はJSPS科研費25330151の助成を受けたものである。

## 参考文献

- [1] Samit, A. and Waters, B.: Fuzzy Identity-Based Encryption, *Proc. 24th Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pp.457–473 (2005).
- [2] Cheung, L. and Newport, C.: Provably Secure Ciphertext Policy ABE, *Proc. 14th ACM Conference on Computer and Communications Security*, pp.456–465 (2007).
- [3] Bethencourt, J., Sahai, A. and Waters, B.: Ciphertext-Policy Attribute-Based Encryption, *Proc. 2007 IEEE Symposium on Security and Privacy*, pp.321–334 (2007).
- [4] Yu, S., Wang, C., Ren, K. and Lou, W.: Attribute Based Data Sharing with Attribute Revocation, *Proc. 5th ACM Symposium on Information, Computer and Communications Security*, pp.261–270 (2010).
- [5] Hur, J. and Noh, D.: Attribute-Based Access Control with Efficient Revocation in Data Outsourcing Systems, *IEEE TPDS*, Vol.22, No.7, pp.1214–1221 (2011).
- [6] Ostrovsky, R., Sahai, A. and Waters, B.: Attribute-Based Encryption with Non-Monotonic Access Structure, *Proc. 14th ACM Conference on Computer and Communications Security*, pp.195–203 (2007).
- [7] Naor, M. and Pinkas, B.: Efficient Trace and Revoke Schemes, *Proc. 4th International Conference on Financial Cryptography*, pp.1–20 (2000).

## 推薦文

本稿は、属性失効機能付き属性ベース暗号の改良についての報告である。既存方式の問題点が精査されており、論文の立脚点が明確であり、不正ユーザ、クラウドサーバ、属性失効ユーザの攻撃に対して安全性検証も適正になされている。今後の同研究分野の発展に有用であるため、推薦論文として推薦したい。

(コンピュータセキュリティ研究会主査 西垣正勝)



成瀬 猛 (学生会員)

平成25年名古屋工業大学工学部情報工学科卒業。同大学大学院工学研究科情報工学専攻博士前期課程在学中。属性ベース暗号の研究に従事。平成25年DICOMO2013優秀プレゼンテーション賞、DICOMO2013優秀論文賞。

文賞。



毛利 公美

平成5年愛媛大学工学部情報工学科卒業。平成7年同大学大学院工学研究科情報工学専攻博士前期課程修了。平成14年博士(工学)(徳島大)。平成7年香川短期大学助手。平成10年徳島大学工学部知能情報工学科助手、平成15年同講師。平成19年岐阜大学総合情報メディアセンター准教授。コンピュータネットワーク、ネットワークセキュリティ、符号・暗号理論等の研究・教育に従事。



白石 善明 (正会員)

平成7年愛媛大学工学部情報工学科卒業。平成9年同大学大学院博士前期課程修了。平成12年徳島大学大学院博士後期課程修了。博士(工学)。平成14年近畿大学理工学部情報学科講師。平成18年名古屋工業大学大学院情報工学専攻助教授。平成25年神戸大学大学院電気電子工学専攻准教授。情報セキュリティ、コンピュータネットワーク、教育支援、知識流通支援等の研究・教育に従事。平成14年電子情報通信学会オフィスシステム研究賞、平成15年暗号と情報セキュリティシンポジウム(SCIS)20周年記念賞、平成18年SCIS論文賞。平成19, 20, 23, 25年DICOMO 2007, 2008, 2011, 2013優秀論文賞。平成24年電子情報通信学会ライフインテリジェンスとオフィス情報システム研究会功労賞。