情報処理学会論文誌 Vol.63 No.9 1419–1439 (Sep. 2022)

標準モデルで安全性証明可能なプロキシ支援による関数型暗号の復号鍵失効機能の実現

松田 規[1,2,a)] 川合 豊[1] 平野 貴人[1] 伊藤 隆[1] 西垣 正勝[2]

受付日 2021年12月7日, 採録日 2022年6月14日

概要：クラウドサーバへ企業機密が保管される機会が増加するにつれ，機密保護のために暗号化が注目されている．特に，用途に応じて柔軟な復号権限を設定できる関数型暗号の活用が有効と考えられる．しかし，異動や退職，復号鍵紛失などによる復号鍵の失効が課題となる．効率的な失効方式として，プロキシ支援型のアプローチが提案されているが，既存研究では主に暗号文ポリシ型を対象とし，安全性も selective 条件下でしか与えられていない．本論文では，関数型暗号の復号鍵を，プロキシ支援の下で効率的に失効させる方式を提案する．提案方式は，復号鍵を要素ごとに分解して 2 分割してプロキシ鍵とユーザ秘密鍵を生成することで，両者が揃ったときだけ復号可能とする．そのため，プロキシサーバが管理するプロキシ鍵を削除するだけで即時にユーザ秘密鍵を失効できる．また，鍵ポリシ型と暗号文ポリシ型の双方で実現可能であることも示す．さらに，攻撃者としてクラウドストレージ，プロキシサーバ，失効ユーザを包含する安全性モデルを定義し，adaptive 状況下で安全性証明を与える．

キーワード：関数型暗号，属性ベース暗号，復号鍵失効

Fully-secure Proxy-assisted Revocable Functional Encryption for Cloud Storage

Nori Matsuda[1,2,a)] Yutaka Kawai[1] Takato Hirano[1] Takashi Ito[1] Masakatsu Nishigaki[2]

Received: December 7, 2021, Accepted: June 14, 2022

Abstract: While secret documents are stored on cloud servers, encryption is focused as one of the solutions to protect their secrecy from the cloud servers. In particular, functional encryption is suitable because of its flexible access control function, However, key revocation function should be required in actual applications. Therefore, a proxy-assisted approach has been proposed as an effective key revocation scheme. Previous researches have mainly focused on the ciphertext-policy attribute-based encryption and security is proved only in the selective adversarial model. In this paper, we propose an efficient and secure key revocation schemes for both the key-policy and ciphertext-policy functional encryption in the proxy assisted approach. In our schemes, a proxy key is generated from the attribute or access policy related part of secret key of functional encryption, and a user private key is generated from the rest of the secret key. Because both the proxy key and user private key is required for decryption, it's enough to delete the proxy key from the proxy server in case of key revocation. In addition, we define a security model that includes cloud storage, proxy servers, and revoked users as an adversary and we prove that our schemes are secure in the adaptive adversarial model.

Keywords: functional encryption, attribute-based encryption, key revocation, proxy assisted approach

[1] 三菱電機株式会社
Mitsubishi Electric Corporation, Kamakura, Kanagawa 247–8501, Japan
[2] 静岡大学創造科学技術大学院
Graduate School of Science and Technology, Shizuoka University, Hamamatsu, Shizuoka 432–8011, Japan
[a)] Matsuda.Nori@ea.MitsubishiElectric.co.jp

1. はじめに

1.1 背景

クラウドサービスの普及により，企業がデータを外部のクラウドストレージに保管する機会が増えている．これら

のデータには企業機密や個人情報などの機微な情報を含むため，データを暗号化してクラウドストレージに保管するのが好ましい．特に，データ暗号化の際に条件を指定でき，条件を満たすユーザのみが暗号化データを復号できる関数型暗号（Functional Encryption）[*1]や属性ベース暗号（Attribute-based Encryption）を用いると，暗号化データに復号権限を設定することができるため，機密度や公開範囲の異なる様々なデータを扱う際に都合がよく，これまでに様々な方式が提案されている [3], [7], [14], [18], [25]．

一方，企業では異動・退職にともなってアクセス権が変化するケースや，復号鍵を紛失するケースなどが考えられるため，復号鍵の失効処理，すなわち “それまで読めていたデータを読めなくする処理” が必要となる．失効について，公開鍵と復号鍵が 1 対 1 に対応する RSA 暗号や ID ベース暗号では様々な方式が提案されている [9], [21], [22], [23], [27]．また，複数の受信者が存在する放送型暗号でも，暗号化するときに失効ユーザを受信者から外す仕組みが研究されている [1], [11], [26]．しかし，関数型暗号や属性ベース暗号の場合，データを復号できるユーザが複数存在しうるが，そのうち特定ユーザだけを失効できなければならない．さらに，過去に復号できていた暗号化データも復号できないようにする必要があり，失効を困難なものとしている．たとえば，暗号化データが “A 部の社員であればだれでも復号可能” という条件（ポリシ）で暗号化されている場合を考える．従来，A 部メンバの A 氏，B 氏，C 氏の 3 人ともに復号できたものが，C 氏が異動になった場合に C 氏の復号鍵だけ失効して，A 氏，B 氏は引き続き復号できるようにしなければならない．このように複数の復号鍵の一部だけを，任意の時点から無効化できることが必要である．

そこで，属性ベース暗号の失効方式についても様々な研究がなされている．そのアプローチは Direct Revocation と Indirect Revocation の 2 つに大別できる [2]．Direct Revocation は，暗号化時に失効ユーザを指定することで，失効ユーザが復号できないように暗号化する方式である．一方，Indirect Revocation は，暗号化時にだれが失効しているかを意識する必要がなく，暗号化データの変換や，失効していないユーザ秘密鍵の更新などにより，失効ユーザが復号できないようにする方式である．どちらの方式が良いかはユースケース次第だが，企業機密の保護・保管というユースケースを考慮すると，暗号化した後に復号鍵が漏洩する可能性にも対応できなければならず，Indirect Revocation のアプローチが適している．

そこで我々は，関数型暗号に対して，次に示す要件を満たす失効機能が必要と考えた．

要件 1：暗号化時に失効されたユーザがだれかを意識しないでよいこと

要件 2：復号できていた暗号化データが復号できなくなること

要件 3：失効にともない，クラウドストレージ上の暗号化データの更新処理が不要もしくは軽微であること

要件 4：失効していないユーザの復号鍵は従来どおり使えること

[*1] 本論文では属性ベース暗号の復号鍵失効について議論するが，Takashima らが提案した論文 [18] の表記に基づいて関数型暗号と記載している．

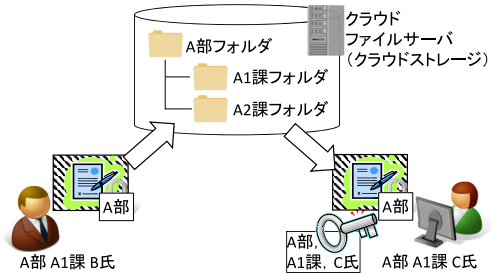


図 1 ファイルサーバのユースケース
Fig. 1 A usecase for cloud file server.

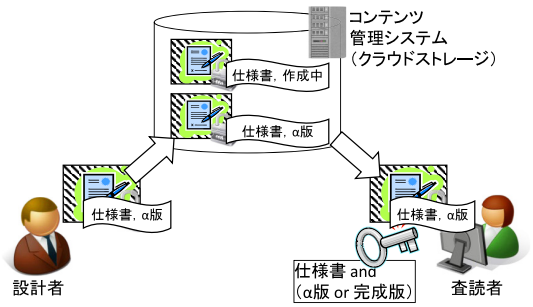


図 2 エンタープライズコンテンツ管理システムのユースケース
Fig. 2 A usecase for enterprise content management system.

1.2 想定ユースケース

企業におけるクラウド活用による情報管理のユースケースについて例を交えて説明する．**図 1** は，一般的なファイルサーバをクラウドストレージ上に移行したユースケースである．クラウドファイルサーバ（クラウドストレージ）上には，部や課の単位でフォルダが作成され，その部や課に所属するユーザにアクセス権限が与えられる．データを作成したユーザ（図中 “A 部 A1 課 B 氏”）は，そのデータをだれと共有するかを判断し，たとえば “A 部” 内で共有したい場合は，“A 部” のフォルダにデータを置く．このようなケースでは，データを暗号化する際には，たとえば “A 部” のようにだれが閲覧できるかというアクセス権限を指定して暗号化を行い，ユーザにはたとえば “A 部，A1 課，C 氏” のような属性を含む復号鍵を配布する暗号文ポリシ型関数型暗号の利用が適している．

別のユースケースとして，**図 2** にエンタープライズコンテンツ管理システム（クラウドストレージ）のユースケースを示す．このユースケースでは，前述のファイルサーバと異なり，クラウドストレージ上のレポジトリにデータが

蓄積される．このとき，たとえば “仕様書，α 版” のように，データの属性を表すタグを付与して保管する．データを閲覧するユーザには，業務内容などを加味して，たとえば “仕様書 and（α 版 or 完成版）” のようにアクセスできるデータの条件がアクセス権限として付与されており，そのアクセス権限に合致する属性のデータを検索・閲覧することができる．このユースケースでは，データに付与するタグを属性として暗号化し，ユーザに配布する復号鍵にはアクセス権限を指定できる鍵ポリシ型関数型暗号の利用が適する．

以上のように，データ管理方法に応じて，暗号文ポリシ型関数型暗号（CP-FE もしくは CP-ABE）と鍵ポリシ型関数型暗号（KP-FE もしくは KP-ABE）を使い分ける必要があることから，鍵ポリシ型と暗号文ポリシ型の両者で復号鍵の失効を実現できることが好ましい．

1.3 関連研究

関数型暗号の復号鍵失効に関して，様々な方式が提案されている．Bethencourt らは，CP-ABE を対象に，復号鍵に有効期限を付けて失効を行う方式を提案している [3]．本方式では，復号鍵に有効期限を入れ，暗号化データには暗号化時刻を入れることで，有効期限が切れた復号鍵では復号ができないようにする．しかし，復号鍵の有効期限が切れても，それまで復号できていた暗号化データは引き続き復号でき，さらに復号鍵を任意のタイミングで失効できないなどの課題があり，要件 2 を満たさない．また，復号鍵の有効期限と暗号化時刻の大小比較ができるようにアクセス構造を設定するとアクセス構造が複雑化しやすく暗号化や復号に時間がかかったり，復号鍵の定期的な更新が必要になったりするなどの課題もあり，要件 4 を満たさない．

Attrapadung らは，ユースケースに応じて Direct Revocation と Indirect Revocation を使い分けられるように，それらを統合した方式を提案している [2]．この方式では，暗号化する際に Direct/Indirect Revocation を指定でき，なおかつユーザは 1 つの秘密鍵でいずれの暗号化データも復号できるようになる．しかし，Direct モードでは，暗号化時に失効されているユーザがだれかを指定しなければならず要件 1 を満たさない．また，Indirect モードでは，暗号化データとユーザ秘密鍵に時刻情報を埋め込み，時刻情報が一致したときだけ復号ができる．失効されていないユーザは秘密鍵を更新して復号できる時刻を延長していくが，過去に復号できていた暗号化データを復号できなくなる仕組みはなく，要件 2 を満たさない．

Yu らは，復号鍵と暗号化データにバージョン情報を埋め込むことで失効を行う方式を提案している [33]．この方式では，失効していない復号鍵のバージョン番号をあげて，さらに暗号化データのバージョン番号を Proxy Re-encryption で更新することで，バージョン番号が更新されていない失効した復号鍵が使えないようにする．しかし，復号鍵の失効が生じるたびに暗号化データを変換する必要があることや，失効していないユーザの復号鍵をすべて更新する必要があることから，要件 3 や要件 4 を満たさない．Li ら [15] は，ユーザが失効するたびにグループ公開鍵を更新し，同時にユーザの復号鍵とクラウドストレージ上の暗号化データを更新することで失効を実現する仕組みを提案しているが，Yu らの方式と同様に暗号化データの変換が必要という課題がある．Wang ら [24] によって，階層型 ID ベース暗号と属性ベース暗号方式を組み合わせた方式も提案されているが，本方式も同様に失効時に暗号化データの変換が必要である．Li らの方式 [16]，Yang らの方式 [29]，Yu らの方式 [32] では，失効される属性に関連する暗号化データに限定して再暗号化を行うように効率化を図ったものの，依然として暗号化データの変換が必要という課題は残ったままである．Sahai ら [20] は，公開情報だけを用いて暗号化データのアクセス権限に制限を加えていく仕組みを提案し，失効ユーザが復号できないようにする Revocable-Storage Attribute-Based Encryption を実現した．しかし，秘密鍵を失効させるために，クラウドストレージ上のすべての暗号化データを確認し，失効した秘密鍵で復号可能な暗号文を変換していく必要があるため要件 3 を満たさない．Lee ら [13] は，暗号化データのアクセス権限に制限を加える変更が可能な Ciphertext Delegatable Encryption，それを用いて復号可能な時刻を制約可能な Self-Updatable Encryption という新しいプリミティブを定義した．これらを用いて，Revocable-Storage Attribute-Based Encryption を構築する方式を提案した．しかし，Sahai らの方式と同様に，失効した秘密鍵で復号可能な暗号文をすべて変換していく必要があり要件 3 を満たさない．Yamada ら [28] は，属性ベース暗号に対して失効機能を付加する構成法として，Pair encoding framework に基づく方式と，ブール式に基づく属性ベース暗号を拡張する方式を提案した．しかし，いずれの方式も暗号化時に失効されたユーザのリストを指定する必要があり，要件 1 や要件 2 を満たさない．森ら [34] は，non-monotone なアクセス構造を持つ CP-ABE を活用して，暗号化データに NOT 条件を付加する復号鍵の失効，および Kawai ら [10] のプロキシ再暗号化による復号鍵の再有効化方式を提案している．本方式では，クラウドストレージ外部に置かれた再暗号化装置で暗号文の再暗号化を実施しているためクラウドストレージの負荷は直接的に増えていないが，復号鍵の失効や再有効化のために暗号化データの再暗号化が必要であり，要件 3 を満たさない．

Yang らは，プロキシ支援の下で効率的に失効を行う方式を提案している [30], [31]．この方式では，通常の復号鍵をプロキシ鍵とユーザ秘密鍵に分割して 2 段階復号を行うようにして，プロキシ鍵を削除して対応するユーザ秘密

鍵を使えなくすることで失効を実現する．本方式は，復号鍵を任意のタイミングで失効でき，クラウドストレージ上の暗号化データを再暗号化する必要もなく，失効されないユーザ秘密鍵を更新する必要がないという利点があり，要件 1，2，3，4 をすべて満たす．しかし，Yang らの方式は CP-ABE でしか実現されておらず，KP-ABE での実現方式は示されていない．さらに安全性証明がジェネリック群モデルであるという制約がついている．Green らは，CP-ABE と KP-ABE の両方について実現方式を提案しているが，安全性証明が selective という制約がある [8]．

また，Nomura らは，Multi-authority 環境の下で，プロキシ支援で失効を実現する方式を提案している [17]．しかし，Yang らの方式と同様に，CP-ABE についてのみ実現しており，KP-ABE での実現方式が示されていない点と，安全性証明が selective であるという制約がついている．Fan らが提案する方式 [6] も，Nomura らの方式と同様に CP-ABE のみ対応しており，安全性証明も selective であるという制約がついている．

1.4 我々の貢献

本論文では，1.1 節で示した要件 1〜4 を満たしつつ，adaptive 状況下で安全性証明が可能な方式を提案する．具体的には，企業内データを外部のクラウドストレージに保管する状況を想定し，人事部や情報システム部門によってユーザ秘密鍵が発行される Single-Authority モデルの関数型暗号が適すると考え，Takashima ら [18] が提案した鍵ポリシ型および暗号文ポリシ型の方式を採用する．これをベースとして，失効の即時性，クラウドストレージの負荷低減に優れ，要件 1〜4 を満たすことができるプロキシ支援型のアプローチに基づいて，失効機能付き鍵ポリシ型関数型暗号と失効機能付き暗号文ポリシ型関数型暗号を実現する．これは，関数型暗号の復号鍵を要素ごとに分解して，プロキシ鍵とユーザ秘密鍵に分割し，プロキシ鍵とユーザ秘密鍵のペアがなければ暗号化データが復号できないようにする．このプロキシ鍵とユーザ秘密鍵のペアをユーザごとに生成し，ユーザにはユーザ秘密鍵だけを配布する．ユーザ秘密鍵を失効させる場合は，対応するプロキシ鍵を削除することで，ユーザ秘密鍵だけでは復号ができないようになり，結果的に失効が実現できる．

また，あらゆる暗号化データを入手できるクラウドストレージ，全ユーザのプロキシ鍵を入手できるプロキシサーバ，失効したユーザ秘密鍵を持つユーザの 3 者を攻撃者と想定し，このいずれの攻撃者であっても暗号化データの識別ができないことを示す安全性モデルを定義する．そして，本モデルに基づいて提案方式が adaptive な攻撃者に対して安全であることを示す．

1.5 本論文の構成

初めに 2 章で，Takashima ら [18] によって提案された DPVS や安全性仮定などの定義について示す．次に 3 章では，想定するシステムモデルと，要件の詳細について説明する．4 章で提案方式である失効機能付き鍵ポリシ型関数型暗号，および失効機能付き暗号文ポリシ型関数型暗号のアルゴリズムを示し，5 章で実システム適用時の鍵運用方法について示す．最後に，6 章で本提案方式が要件を満たすことの考察を行う．安全性証明は付録で示す．

2. 準備

Takashima ら [18] による DPVS や安全性仮定について示す．

2.1 表記方法

A を確率変数としたとき，確率変数 A の分布に従ってランダムに要素 y を選ぶことを $y \xleftarrow{\mathsf{R}} A$ と書く．A が集合のとき，一様分布で要素 y を選ぶことを $y \xleftarrow{\mathsf{U}} A$ と書く．y を z によって定義するとき，$y := z$ と書く．値 a を何らかの定数としたとき，アルゴリズム A が入力 x によって値 a を出力することを $A(x) \to a$ と書く．任意の多項式 $f(x)$ およびセキュリティパラメータ λ に対して，確率 p が $1/f(\lambda)$ より小さいとき，確率が negligible であるという．

素数 q が与えられたとき，位数 q の有限体を $\mathbb{F}_q$ と記す．また，有限体 $\mathbb{F}_q$ から零元を取り除いたものを $\mathbb{F}_q^\times := \mathbb{F}_q \backslash \{0\}$ と記す．有限体 $\mathbb{F}_q$ 上で定義された n 次元ベクトル $(x_1, \ldots, x_n) \in \mathbb{F}_q^n$ を $\overrightarrow{x}$ と記す．特別にすべての要素が 0 からなる n 次元ベクトルを $\overrightarrow{0}$，すべての要素が 1 からなる n 次元ベクトルを $\overrightarrow{1}$ と記す．2 つのベクトル $\overrightarrow{x} = (x_1, \ldots, x_n)$，$\overrightarrow{v} = (v_1, \ldots, v_n)$ の内積 $\sum_{i=1}^n x_i v_i$ を $\overrightarrow{x} \cdot \overrightarrow{v}$ と記す．行列 X の転置行列を X^T と記す．また，$\ell \times \ell$ の単位行列を I_ℓ と記す．ベクトル空間 $\mathbb{V}$ の要素を太字で $\boldsymbol{x} \in \mathbb{V}$ と記す．ベクトル $\boldsymbol{b}_i \in \mathbb{V}$ $(i = 1, \ldots, n)$ が与えられたとき，それらのベクトルで張られる部分空間のことを $\mathsf{span}\langle \boldsymbol{b}_1, \ldots, \boldsymbol{b}_n \rangle \subseteq \mathbb{V}$ と記す．また，ベクトル $\overrightarrow{x} = (x_1, \ldots, x_n)$ と基底 $\mathbb{B} := (\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n)$ が与えられたとき，ベクトルの線形和 $\sum_{i=1}^n x_i \boldsymbol{b}_i$ を $(\overrightarrow{x})_{\mathbb{B}} = (x_1, \ldots, x_n)_{\mathbb{B}}$ と記す．属性フォーマット $\overrightarrow{n} := (d; n_1, \ldots, n_d)$ が与えられたとき，すべての $t = 1, \ldots, d$ に対して標準基底 $(\overbrace{0, \ldots, 0}^{j-1}, 1, \overbrace{0, \ldots, 0}^{n_t - j})$ を $\overrightarrow{e}_{t,j}$ と記す．

2.2 Dual Pairing Vector Spaces（DPVS）

関数型暗号の基礎となる DPVS について，その定義と性質を述べる．初めに，対称ペアリング群の定義を示す．

Definition 1. 対称ペアリング群 $(q, \mathbb{G}, \mathbb{G}_T, g, e)$ は，素数 q，位数 q の有限巡回乗法群 $\mathbb{G}$ と $\mathbb{G}_T$，群 $\mathbb{G}$ の生成元 g，多項式時間で計算可能な非退化ペアリング写像 $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ か

ら構成される．ここで，$e(g^s, g^t) = e(g,g)^{st}$ かつ $e(g,g) \neq 1$ が成り立つ．アルゴリズム $\mathcal{G}_{\mathsf{bpg}}$ を，セキュリティパラメータ 1^λ を入力として受け取り，対称ペアリング群 $(q, \mathbb{G}, \mathbb{G}_T, g, e)$ を生成する関数として定義する．

次に，対称ペアリング群を用いたDPVSの定義を示す．

Definition 2. 対称ペアリング群 $(q, \mathbb{G}, \mathbb{G}_T, g, e)$ の直積で構成される *"Dual pairing vector spaces (DPVS)"* $(q, \mathbb{V}, \mathbb{G}_T, \mathbb{A}, e)$ は，素数 q，$\mathbb{F}_q$ 上で定義される N 次元ベクトル空間 $\mathbb{V} := \overbrace{\mathbb{G} \times \cdots \times \mathbb{G}}^{N}$，位数 q の巡回群 $\mathbb{G}_T$，N 次元ベクトル空間 $\mathbb{V}$ の標準基底 $\mathbb{A} := (\boldsymbol{a}_1, \ldots, \boldsymbol{a}_N)$，ペアリング写像 $e : \mathbb{V} \times \mathbb{V} \to \mathbb{G}_T$ で構成される．ここで，$\boldsymbol{a}_i := (\overbrace{1, \ldots, 1}^{i-1}, g, \overbrace{1, \ldots, 1}^{N-i})$ である．

ペアリング写像は，2つのベクトル $\boldsymbol{x} := (g_1, \ldots, g_N) \in \mathbb{V}$ と $\boldsymbol{y} := (h_1, \ldots, h_N) \in \mathbb{V}$ があったとき，$e(\boldsymbol{x}, \boldsymbol{y}) := \prod_{i=1}^{N} e(g_i, h_i) \in \mathbb{G}_T$ と定義する．このペアリング写像は非退化写像であり，$e(s\boldsymbol{x}, t\boldsymbol{y}) = e(\boldsymbol{x}, \boldsymbol{y})^{st}$ となる．また，すべての $\boldsymbol{y} \in \mathbb{V}$ について $e(\boldsymbol{x}, \boldsymbol{y}) = 1$ が成り立つなら，$\boldsymbol{x} = \boldsymbol{0}$ となる．また，$g_T := e(g,g) \neq 1 \in \mathbb{G}_T$ であり，すべての i，j に対して $e(\boldsymbol{a}_i, \boldsymbol{a}_j) = g_T^{\delta_{i,j}}$ が成り立つ．ここで，$\delta_{i,j}$ はクロネッカーのデルタである．

DPVS 生成アルゴリズム $\mathcal{G}_{\mathsf{dpvs}}$ は，セキュリティパラメータ 1^λ，次元 $N \in \mathbb{N}$ を入力として受け取り，$\mathbf{param}_{\mathbb{V}} := (q, \mathbb{V}, \mathbb{G}_T, \mathbb{A}, e)$ を生成する．

関数型暗号では，DPVS上で定義されるランダム双対直交基底を用いるので，その生成アルゴリズムの定義を示す．

Definition 3. *"Random dual orthonormal bases generator"* $\mathcal{G}_{\mathsf{ob}}$ は，下記のように定義される．

$\mathcal{G}_{\mathsf{ob}}(1^\lambda, \overrightarrow{n} := (d; n_1, \ldots, n_d)) :$

$\mathbf{param}_{\mathbb{G}} := (q, \mathbb{G}, \mathbb{G}_T, g, e) \xleftarrow{\mathsf{R}} \mathcal{G}_{\mathsf{bpg}}(1^\lambda)$,

$\psi \xleftarrow{\mathsf{U}} \mathbb{F}_q^\times, N_0 := 5, N_t := 3n_t + 1$ for $t = 1, \ldots, d$,

for $t = 0, \ldots, d$;

$\mathbf{param}_{\mathbb{V}_t} \xleftarrow{\mathsf{R}} \mathcal{G}_{\mathsf{dpvs}}(1^\lambda, N_t, \mathbf{param}_{\mathbb{G}})$,

$$X_t := \begin{pmatrix} \overrightarrow{\chi}_{t,1} \\ \vdots \\ \overrightarrow{\chi}_{t,N_t} \end{pmatrix} := (\chi_{t,i,j})_{i,j} \xleftarrow{\mathsf{U}} GL(N_t, \mathbb{F}_q),$$

$$\begin{pmatrix} \overrightarrow{\vartheta}_{t,1} \\ \vdots \\ \overrightarrow{\vartheta}_{t,N_t} \end{pmatrix} := (\vartheta_{t,i,j})_{i,j} := \psi \cdot (X_t^T)^{-1},$$

$$\boldsymbol{b}_{t,i} := (\overrightarrow{\chi}_{t,i})_{\mathbb{A}_t} = \sum_{j=1}^{N_t} \chi_{t,i,j} \boldsymbol{a}_{t,j} \text{ for } i = 1, \ldots, N_t,$$

$$\boldsymbol{b}^*_{t,i} := (\overrightarrow{\vartheta}_{t,i})_{\mathbb{A}_t} = \sum_{j=1}^{N_t} \vartheta_{t,i,j} \boldsymbol{a}_{t,j} \text{ for } i = 1, \ldots, N_t,$$

$\mathbb{B}_t := (\boldsymbol{b}_{t,1}, \ldots, \boldsymbol{b}_{t,N_t}), \mathbb{B}^*_t := (\boldsymbol{b}^*_{t,1}, \ldots, \boldsymbol{b}^*_{t,N_t})$,

$g_T := e(g,g)^\psi, \mathbf{param}_{\overrightarrow{n}} := (\{\mathbf{param}_{\mathbb{V}_t}\}_{t=0,\ldots,d}, g_T)$,

return $(\mathbf{param}_{\overrightarrow{n}}, \{\mathbb{B}_t, \mathbb{B}^*_t\}_{t=0,\ldots,d})$.

なお，すべての $t = 0, \ldots, d;\ i = 1, \ldots, N_t$ に対して，$e(\boldsymbol{b}_{t,i}, \boldsymbol{b}^*_{t,i}) = g_T$ となる．

2.3 スパンプログラムとアクセス構造

アクセス構造の表現のために non-monotone なスパンプログラムを用いる．その定義を下記に示す．

Definition 4. $\{p_1, \ldots, p_n\}$ を変数の集合とする．このとき，有限体 $\mathbb{F}_q$ 上で定義されるスパンプログラムは，$\mathbb{F}_q$ 上の $(\ell \times r)$ 行列 M を用いて $\hat{M} := (M, \rho)$ と定義される．ここで，ρ はラベリングを行う写像であり，行列 M の行と，集合 $\{p_1, \ldots, p_n, \neg p_1, \ldots, \neg p_n\}$ の要素との対応をとる．

また，入力列 $\delta \in \{0,1\}^n$ に対して，行列 M の部分行列 M_δ を定める．これは，行列 M の各 j 行目の要素 M_j に対して，$\rho(j) = p_i$ かつ $\delta_i = 1$，もしくは $\rho(j) = \neg p_i$ かつ $\delta_i = 0$ となる場合を $\gamma(j) = 1$ と定義し，$M_\delta := (M_j)_{\gamma(j)=1}$ で得ることができる．

スパンプログラム $\hat{M}$ が δ を受け入れるとは，$\overrightarrow{1} \in \mathsf{span}\langle M_\delta \rangle$ が成り立つとき，ただそのときだけである．スパンプログラムを用いることで，入力 δ を受け入れるときのみ $f(\delta) = 1$ となるブール関数を構築することができる．なお，ラベリング写像 ρ が集合 $\{p_1, \ldots, p_n\}$ にしか写像しない場合，スパンプログラムは *monotone* であるという．一方，集合 $\{p_1, \ldots, p_n, \neg p_1, \ldots, \neg p_n\}$ に写像される場合，*non-monotone* であるという．

このスパンプログラムを用いて，属性がベクトル表現で表されるときのアクセス構造を下記のように定義する．本論文では，non-monotone なスパンプログラムを用いる．

Definition 5. アクセス構造 $\mathbb{S}$ はスパンプログラム $\hat{M} := (M, \rho)$ であり，変数 p は $t \in \{1, \ldots, d\}$，$\overrightarrow{v} \in \mathbb{F}_q^{n_t} \backslash \{\overrightarrow{0}\}$ のとき $(t, \overrightarrow{v})$ で表され，写像 ρ は $\rho : \{1, \ldots, \ell\} \to \{(t_1, \overrightarrow{v_1}), (t_2, \overrightarrow{v}_2), \ldots, \neg(t_1, \overrightarrow{v}_1), \neg(t_2, \overrightarrow{v}_2), \ldots\}$ で定義される．

属性集合 Γ は，集合 $\{1, \ldots, d\}$ の部分集合の要素 t を用いて，$\Gamma := \{(t, \overrightarrow{x}_t) | \overrightarrow{x}_t \in \mathbb{F}_q^{n_t} \backslash \{\overrightarrow{0}\}, 1 \leq t \leq d\}$ と表される．

属性集合 Γ とアクセス構造 $\mathbb{S}$ が与えられたとき，スパンプログラム $\hat{M} := (M, \rho)$ に対する写像 γ を次のように定義する．$i = 1, \ldots, \ell$ に対して，もし $[\rho(i) = (t, \overrightarrow{v}_i)] \wedge [(t, \overrightarrow{x}_t) \in \Gamma] \wedge [\overrightarrow{v}_i \cdot \overrightarrow{x}_t = 0]$ または $[\rho(i) = \neg(t, \overrightarrow{v}_i)] \wedge [(t, \overrightarrow{x}_t) \in \Gamma] \wedge [\overrightarrow{v}_i \cdot \overrightarrow{x}_t \neq 0]$ が成り立つ場合，$\gamma(i) = 1$ とし，それ以外の場合は $\gamma(i) = 0$ とする．

アクセス構造 $\mathbb{S} := (M, \rho)$ が属性集合 Γ を受け入れるとは，$\overrightarrow{1} \in \mathsf{span}\langle (M_i)_{\gamma(i)=1} \rangle$ が成り立つときを指す．

このスパンプログラムを用いて，秘密分散法を定義する．

Definition 6. スパンプログラム $\hat{M} := (M, \rho)$ に対する秘密分散法は下記のように構成される．

(1) M を $\ell \times r$ 行列とする．また，$\overrightarrow{f}^T := (f_1, \ldots, f_r)^T \xleftarrow{\mathsf{U}} \mathbb{F}_q^r$ とする．$s_0 := \overrightarrow{1} \cdot \overrightarrow{f}^T = \sum_{k=1}^{r} f_k$ を分散したい秘密値としたとき，$\overrightarrow{s}^T := (s_1, \cdots, s_\ell)^T := M \cdot \overrightarrow{f}^T$ は ℓ 個の秘密分散値からなるベクトルである．また，各秘密分散値 s_i は $\rho(i)$ と対応する．

(2) アクセス構造 $\mathbb{S}$ が属性集合 Γ を受け入れる場合，すなわち $\gamma : \{1, \ldots, \ell\} \to \{0,1\}$ に対して $\overrightarrow{1} \in \mathsf{span}\langle (M_i)_{\gamma(i)=1} \rangle$ が成り立つ場合，$I \subseteq \{i \in \{1, \ldots, \ell\} | \gamma(i) = 1\}$ かつ $\sum_{i \in I} \alpha_i s_i = s_0$ となる定数 $\{\alpha_i \in \mathbb{F}_q | i \in I\}$ が存在する．

2.4 安全性仮定

初めに，安全性を帰着させる DLIN 仮定について述べる．

Definition 7. *Decisional Linear Assumption*（*DLIN* 仮定）とは，$(\mathsf{param}_{\mathbb{G}}, g, g^{\xi}, g^{\kappa}, g^{\delta\xi}, g^{\sigma\kappa}, Y_\beta) \xleftarrow{\mathsf{R}} \mathcal{G}_\beta^{\mathsf{DLIN}}(1^\lambda)$ が与えられたとき，$\beta \in \{0,1\}$ を推測する問題である．ここで，$\beta \xleftarrow{\mathsf{U}} \{0,1\}$ に対して，

$\mathcal{G}_\beta^{\mathsf{DLIN}}(1^\lambda)$:
$\mathsf{param}_{\mathbb{G}} := (q, \mathbb{G}, \mathbb{G}_T, g, e) \xleftarrow{\mathsf{R}} \mathcal{G}_{\mathsf{bpg}}(1^\lambda)$,
$\kappa, \delta, \xi, \sigma \xleftarrow{\mathsf{U}} \mathbb{F}_q, Y_0 := g^{(\delta+\sigma)}, Y_1 \xleftarrow{\mathsf{U}} \mathbb{G}$,
return $(\mathsf{param}_{\mathbb{G}}, g, g^{\xi}, g^{\kappa}, g^{\delta\xi}, g^{\sigma\kappa}, Y_\beta)$.

確率的アルゴリズム $\mathcal{E}$ に対して，$\mathcal{E}$ のアドバンテージを下記のように定義する．

$$\mathsf{Adv}_{\mathcal{E}}^{\mathsf{DLIN}}(\lambda) := \left| \mathsf{Pr}\left[\mathcal{E}(1^\lambda, \varrho) \to 1 \middle| \varrho \xleftarrow{\mathsf{R}} \mathcal{G}_0^{\mathsf{DLIN}}(1^\lambda) \right] - \mathsf{Pr}\left[\mathcal{E}(1^\lambda, \varrho) \to 1 \middle| \varrho \xleftarrow{\mathsf{R}} \mathcal{G}_1^{\mathsf{DLIN}}(1^\lambda) \right] \right|$$

ここで，確率的アルゴリズム $\mathcal{E}$ が多項式時間アルゴリズムの場合，アドバンテージ $\mathsf{Adv}_{\mathcal{E}}^{\mathsf{DLIN}}(\lambda)$ は *negligible* となる．

失効機能付き関数型暗号では，Desicional Liniear Assumption に帰着可能な下記 2 つの安全性仮定をもちいて安全性を証明する．その定義を示す．

Definition 8. *Problem 1* は，$(\mathbf{param}_{\overrightarrow{n}}, \mathbb{B}_0, \hat{\mathbb{B}_0^*}, \boldsymbol{e}_{\beta,0}, \{\mathbb{B}_t, \hat{\mathbb{B}_t^*}, \boldsymbol{e}_{\beta,t,1}, \boldsymbol{e}_{t,i}\}_{t=1,\ldots,d; i=2,\ldots,n_t}) \xleftarrow{\mathsf{R}} \mathcal{G}_\beta^{\mathsf{P1}}(1^\lambda, \overrightarrow{n})$ が与えられたときに，$\beta \in \{0,1\}$ を推測する問題である．ここで，$\beta \xleftarrow{\mathsf{U}} \{0,1\}$ に対して，

$\mathcal{G}_\beta^{\mathsf{P1}}(1^\lambda, \overrightarrow{n})$:
$(\mathbf{param}_{\overrightarrow{n}}, \{\mathbb{B}_t, \mathbb{B}_t^*\}_{t=0,\ldots,d}) \xleftarrow{\mathsf{R}} \mathcal{G}_{\mathsf{ob}}(1^\lambda, \overrightarrow{n})$,
$\hat{\mathbb{B}_0^*} := (\boldsymbol{b}_{0,1}^*, \boldsymbol{b}_{0,3}^*, \ldots, \boldsymbol{b}_{0,5}^*)$,
$\hat{\mathbb{B}_t^*} := (\boldsymbol{b}_{t,1}^*, \ldots, \boldsymbol{b}_{t,n_t}^*, \boldsymbol{b}_{t,2n_t+1}^*, \ldots, \boldsymbol{b}_{t,3n_t+1}^*)$
for $t = 1, \ldots, d$,
$\omega, z_0, \gamma_0 \xleftarrow{\mathsf{U}} \mathbb{F}_q$,
$\boldsymbol{e}_{0,0} := (\omega, 0, 0, 0, \gamma_0)_{\mathbb{B}_0}, \boldsymbol{e}_{1,0} := (\omega, z_0, 0, 0, \gamma_0)_{\mathbb{B}_0}$,
for $t = 1, \ldots, d$;
$\overrightarrow{e}_{t,1} := (1, 0^{n_t - 1}) \in \mathbb{F}_q^{n_t}, \overrightarrow{z}_t \xleftarrow{\mathsf{U}} \mathbb{F}_q^{n_t}, \gamma_t \xleftarrow{\mathsf{U}} \mathbb{F}_q$,
$\boldsymbol{e}_{0,t,1} := (\omega \overrightarrow{e}_{t,1}, 0^{n_t}, 0^{n_t}, \gamma_t)_{\mathbb{B}_t}$,
$\boldsymbol{e}_{1,t,1} := (\omega \overrightarrow{e}_{t,1}, \overrightarrow{z}_t, 0^{n_t}, \gamma_t)_{\mathbb{B}_t}$,
$\boldsymbol{e}_{t,i} := \omega \boldsymbol{b}_{t,i}$ for $i = 2, \ldots, n_t$,
return $(\mathbf{param}_{\overrightarrow{n}}, \mathbb{B}_0, \hat{\mathbb{B}_0^*}, \boldsymbol{e}_{\beta,0}, \{\mathbb{B}_t, \hat{\mathbb{B}_t^*}, \boldsymbol{e}_{\beta,t,1}, \boldsymbol{e}_{t,i}\}_{t=1,\ldots,d; i=2,\ldots,n_t})$.

確率的アルゴリズム $\mathcal{B}$ に対して，$\mathcal{B}$ のアドバンテージを下記のように定義する．

$$\mathsf{Adv}_{\mathcal{B}}^{\mathsf{P1}}(\lambda) := \left| \mathsf{Pr}\left[\mathcal{B}(1^\lambda, \varrho) \to 1 \middle| \varrho \xleftarrow{\mathsf{R}} \mathcal{G}_0^{\mathsf{P1}}(1^\lambda, \overrightarrow{n}) \right] - \mathsf{Pr}\left[\mathcal{B}(1^\lambda, \varrho) \to 1 \middle| \varrho \xleftarrow{\mathsf{R}} \mathcal{G}_1^{\mathsf{P1}}(1^\lambda, \overrightarrow{n}) \right] \right|$$

Lemma 1. *Problem 1* に対する確率的多項式時間攻撃者 $\mathcal{B}$ が存在するならば，*DLIN* 仮定に対する確率的多項式時間アルゴリズム $\mathcal{E}$ が存在する．攻撃者 $\mathcal{B}$ のアドバンテージは，任意のセキュリティパラメータ λ に対して $\mathsf{Adv}_{\mathcal{B}}^{\mathsf{P1}}(\lambda) \leq \mathsf{Adv}_{\mathcal{E}}^{\mathsf{DLIN}}(\lambda) + (d+6)/q$ となる．

Definition 9. *Problem 2* は，$(\mathbf{param}_{\overrightarrow{n}}, \hat{\mathbb{B}_0}, \mathbb{B}_0^*, \boldsymbol{h}_{\beta,0}^*, \boldsymbol{e}_0, \{\hat{\mathbb{B}_t}, \mathbb{B}_t^*, \boldsymbol{h}_{\beta,t,i}, \boldsymbol{e}_{t,i}\}_{t=1,\ldots,d; i=1,\ldots,n_t}) \xleftarrow{\mathsf{R}} \mathcal{G}_\beta^{\mathsf{P2}}(1^\lambda, \overrightarrow{n})$ が与えられたときに，$\beta \in \{0,1\}$ を推測する問題である．ここで，$\beta \xleftarrow{\mathsf{U}} \{0,1\}$ に対して，

$\mathcal{G}_\beta^{\mathsf{P2}}(1^\lambda, \overrightarrow{n})$:
$(\mathbf{param}_{\overrightarrow{n}}, \{\mathbb{B}_t, \mathbb{B}_t^*\}_{t=0,\ldots,d}) \xleftarrow{\mathsf{R}} \mathcal{G}_{\mathsf{ob}}(1^\lambda, \overrightarrow{n})$,
$\hat{\mathbb{B}_0} := (\boldsymbol{b}_{0,1}, \boldsymbol{b}_{0,3}, \ldots, \boldsymbol{b}_{0,5})$,
$\hat{\mathbb{B}_t} := (\boldsymbol{b}_{t,1}, \ldots, \boldsymbol{b}_{t,n_t}, \boldsymbol{b}_{t,2n_t+1}, \ldots, \boldsymbol{b}_{t,3n_t+1})$
for $t = 1, \ldots, d$,
$\delta, \delta_0, \omega \xleftarrow{\mathsf{U}} \mathbb{F}_q, \tau, u_0 \xleftarrow{\mathsf{U}} \mathbb{F}_q^\times, z_0 := u_0^{-1}$,
$\begin{pmatrix} \overrightarrow{z}_{t,1} \\ \vdots \\ \overrightarrow{z}_{t,n_t} \end{pmatrix} := Z_t \xleftarrow{\mathsf{U}} GL(n_t, \mathbb{F}_q)$ for $t = 1, \ldots, d$,
$\begin{pmatrix} \overrightarrow{u}_{t,1} \\ \vdots \\ \overrightarrow{u}_{t,n_t} \end{pmatrix} := (Z_t^{-1})^T$ for $t = 1, \ldots, d$,
$\boldsymbol{h}_{0,0} := (\delta, 0, 0, \delta_0, 0)_{\mathbb{B}_0^*}, \boldsymbol{h}_{1,0} := (\delta, u_0, 0, \delta_0, 0)_{\mathbb{B}_0^*}$,
$\boldsymbol{e}_0 := (\omega, \tau z_0, 0, 0, 0)_{\mathbb{B}_0}$,
for $t = 1, \ldots, d; i = 1, \ldots, n_t$;
$\overrightarrow{e}_{t,i} := (0^{i-1}, 1, 0^{n_t - i}) \in \mathbb{F}_q^{n_t}, \overrightarrow{\delta}_{t,i} \xleftarrow{\mathsf{U}} \mathbb{F}_q^{n_t}$,
$\boldsymbol{h}_{0,t,i}^* := (\delta \overrightarrow{e}_{t,i}, 0^{n_t}, \delta_{t,i}, 0)_{\mathbb{B}_t^*}$,
$\boldsymbol{h}_{1,t,i}^* := (\delta \overrightarrow{e}_{t,i}, \overrightarrow{u}_{t,i}, \delta_{t,i}, 0)_{\mathbb{B}_t^*}$,
$\boldsymbol{e}_{t,i} := (\omega \overrightarrow{e}_{t,i}, \tau \overrightarrow{z}_{t,i}, 0^{n_t}, 0)_{\mathbb{B}_t}$,
return $(\mathbf{param}_{\overrightarrow{n}}, \hat{\mathbb{B}_0}, \mathbb{B}_0^*, \boldsymbol{h}_{\beta,0}^*, \boldsymbol{e}_0, \{\hat{\mathbb{B}_t}, \mathbb{B}_t^*, \boldsymbol{h}_{\beta,t,i}, \boldsymbol{e}_{t,i}\}_{t=1,\ldots,d; i=1,\ldots,n_t})$.

確率的アルゴリズム $\mathcal{B}$ に対して，$\mathcal{B}$ のアドバンテージを下

記のように定義する．

$$\mathsf{Adv}^{\mathsf{P2}}_{\mathcal{B}}(\lambda) := \left|\mathsf{Pr}\left[\mathcal{B}(1^\lambda,\varrho)\to 1 \middle|\, \varrho \xleftarrow{\mathsf{R}} \mathcal{G}^{\mathsf{P2}}_0(1^\lambda,\vec{n})\right] - \mathsf{Pr}\left[\mathcal{B}(1^\lambda,\varrho)\to 1 \middle|\, \varrho \xleftarrow{\mathsf{R}} \mathcal{G}^{\mathsf{P2}}_1(1^\lambda,\vec{n})\right]\right|$$

Lemma 2. *Problem 2* に対する確率的多項式時間攻撃者 $\mathcal{B}$ が存在するならば，*DLIN* 仮定に対する確率的多項式時間アルゴリズム $\mathcal{E}$ が存在する．攻撃者 $\mathcal{B}$ のアドバンテージは，任意のセキュリティパラメータ λ に対して $\mathsf{Adv}^{\mathsf{P2}}_{\mathcal{B}}(\lambda) \le \mathsf{Adv}^{\mathsf{DLIN}}_{\mathcal{E}}(\lambda) + 5/q$ となる．

3. システムモデルと機能要件

本提案方式では，1.4 節で述べたように，プロキシ支援の下で失効を実現する．そのシステム構成は**図 3** に示す構成となり，下記に示すエンティティで構成される．

- **クラウドストレージ**：暗号化データが保管されるストレージ．クラウドベンダによって運用・管理され，企業がサービス利用契約を締結して利用する．他社によって運営されることから，クラウドストレージは semi-honest であると仮定する．すなわち，正しくデータは保管されるが，クラウドストレージの管理者によってデータ閲覧されるリスクが存在する．
- **鍵管理サーバ**：ユーザの属性や役割に基づき，ユーザごとに異なるプロキシ鍵とユーザ秘密鍵を発行するサーバ．鍵生成や失効管理などセキュリティの要となることから，信頼できるエンティティでなければならない．
- **プロキシサーバ**：鍵管理サーバによってユーザごとに生成されたプロキシ鍵をユーザと紐づけて管理し，ユーザの要求に基づき暗号化データの部分復号を行うサーバ．鍵管理サーバからの指示により，プロキシ鍵の削除も行う．企業内に設置されるケースや，クラウドストレージの付随サービスとして提供されるケースが考えられるため，クラウドストレージと同様に semi-honest と仮定する．
- **作成者**：データの作成者であり，作成したデータを関数型暗号で暗号化して，クラウドストレージに保管するユーザ．次に示す復号者も兼ねることがある．
- **復号者**：ユーザ秘密鍵を保有し，暗号化データを閲覧するユーザ．鍵管理サーバでアクセス権限や属性が管理されている．

ここで，上記のクラウドストレージ，鍵管理サーバ，プロキシサーバは，他エンティティと結託しないと仮定する．

上記システムにおいて，秘匿性を保証することに加え，失効機能として以下の要件を満たす必要がある．

要件 1：暗号化時に失効されたユーザがだれかを意識しないでよいこと．すなわち，暗号化時は従来方式どおりにアクセス可能者の条件のみを指定すればよく，だれが失効されているかを意識する必要がないこと．

要件 2：復号できていた暗号化データが復号できなくなること．すなわち，人事異動などで属性やアクセス権限が変わった際に，またユーザ秘密鍵が漏洩した際に，失効させたユーザ秘密鍵で暗号化データが復号できてはならない．

要件 3：失効にともない，クラウドストレージ上の暗号化データの更新処理が不要もしくは軽微であること．すなわち，クラウドストレージに保管された暗号化データのアクセス構造や属性の付け替えや再暗号化など，失効にともなって追加の処理が発生しないこと．

要件 4：失効していないユーザ秘密鍵は従来どおり使えること．すなわち，失効していないユーザに対して，ユーザ秘密鍵の更新などが不要であること．

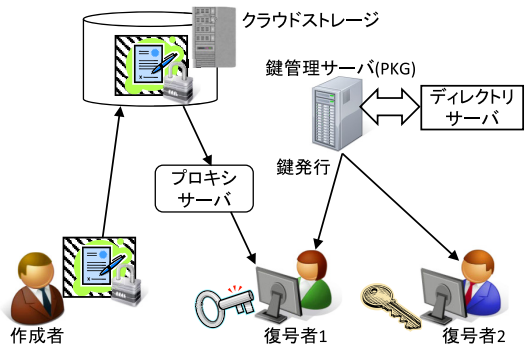


図 3 システムの登場人物
Fig. 3 Entities in our system.

4. 提案方式

失効機能付き鍵ポリシ型関数型暗号，および失効機能付き暗号文ポリシ型関数型暗号の実現方式を提案する．

4.1 実現のためのアイデア

関数型暗号に失効機能を付加するためのアイデアを**図 4** を用いて説明する．関数型暗号では，鍵ポリシ型の場合はアクセス構造 $\mathbb{S}$ を，暗号文ポリシ型の場合はユーザの属性集合 Γ を指定して，復号鍵を生成する．この復号鍵を持つ

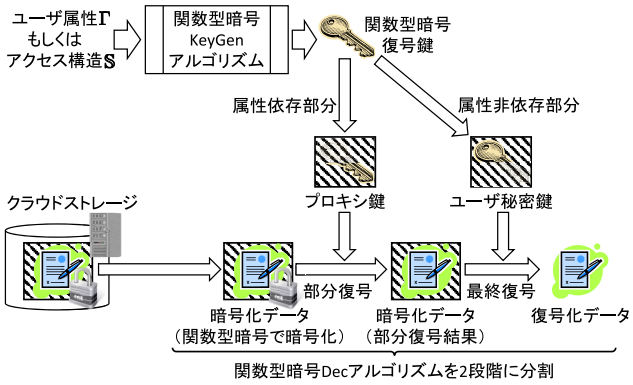


図 4 失効機能付き関数型暗号の実現アイデア
Fig. 4 The idea of revocation mechanism.

ユーザは，暗号化データの復号が可能となる．我々の提案方式では，関数型暗号の復号鍵をユーザごとに生成し，プロキシサーバ用のプロキシ鍵と，ユーザ用のユーザ秘密鍵の 2 個に分割する．この構成より，暗号化データを復号する際は，最初にプロキシ鍵で暗号化データを部分復号するが，復号処理が途中で止まった状態となる．その部分復号結果を，部分復号に用いたプロキシ鍵と対になるユーザ秘密鍵を用いて最終復号することで，復号結果であるデータが得られる．そのため，プロキシ鍵を削除することで，対応するユーザ秘密鍵だけでは暗号化データが復号できなくなり，ユーザ秘密鍵の失効が実現できる．

また，暗号化データの秘匿性に加えて失効機能の有効性を示すためには，暗号化データを管理するクラウドストレージ，全ユーザのプロキシ鍵を管理するプロキシサーバ，失効したユーザ秘密鍵を持つユーザの 3 者を攻撃者と想定し，いずれの攻撃者も暗号化データからいっさいの情報が得られないことを示す必要がある．そこで安全性モデルとして，チャレンジ暗号文を復号できない場合はプロキシ鍵とユーザ秘密鍵の双方をクエリできるのに加え，復号できる場合でもプロキシ鍵かユーザ秘密鍵の一方をクエリすることを許容した状況で，攻撃者がチャレンジ暗号文の識別ができないことをモデル化した．そして，本モデルに基づいて提案方式が adaptive な攻撃者に対して安全であることを示す．

ここで課題となるのは，Dual system encryption に基づいて安全性証明がなされた方式では，チャレンジ暗号文を復号できる属性やアクセス構造を用いて復号鍵をクエリすると，semi-functional な復号鍵が semi-functional チャレンジ暗号文と correlation を起こす点にある [5], [12]．関数型暗号では，自明な攻撃を防ぐためにチャレンジ暗号文を復号できる復号鍵をクエリできないという制約を持たせており，この correlation は問題にならなかった．しかし，本提案方式では，チャレンジ暗号文を復号できる属性やアクセス構造であっても，プロキシ鍵かユーザ秘密鍵の一方であればクエリを許すため，この点が問題となる．そこで我々は，復号鍵を要素ごとに分解し，correlation を起こす要素をプロキシ鍵とユーザ秘密鍵に分けて配置することで，チャレンジ暗号文を復号できる属性やアクセス構造を指定された場合でも，いずれか一方しか入手できない攻撃者のビューからは correlation が見えないような仕組みを実現した．これにより，Dual system encryption の枠組みのままで安全性証明を可能とした．

なお，Datta ら [4] によって，Dual system encryption の枠組みで安全性証明がなされた属性ベース暗号 [19] に対して SD 法のアイデアを適用して direct revocation による失効を実現し，adaptive 条件下で安全性証明を付与した方式が提案されている．この方式でも，チャレンジ暗号文生成時に指定された失効リストに掲載されたユーザ ID を指定して復号鍵をクエリするという制約の下であれば，チャレンジ暗号文の属性 Γ^* を満たすアクセス構造 $\mathbb{S}$ が指定された場合でも復号鍵をシミュレートでき，チャレンジ暗号文と復号鍵の correlation を攻撃者に対して隠すことができると示されている．しかし，彼らのアプローチは，チャレンジ暗号文と復号鍵で correlation を起こす乱数の一部を，SD 法で生成した covering set と private set でそれぞれ暗号化することで correlation を隠す方式である．そのため，暗号化時に失効ユーザが確定する direct revocation であれば有効なテクニックであるが，我々のような暗号化時に失効ユーザが確定しない indirect revocation による失効方式に適用することができない．我々も correlation が生じる要素を攻撃者に対して隠すという考え方は同じだが，indirect revocation に適用可能な鍵分割による correlation の隠し方を実現した．

4.2 失効機能付き鍵ポリシ型関数型暗号

鍵ポリシ型関数型暗号に対して失効機能を付加したアルゴリズム，およびその安全性について示す．

4.2.1 アルゴリズム定義とセキュリティ定義

失効機能付き鍵ポリシ型関数型暗号（RKP-FE）のアルゴリズム定義を下記に示す．

Definition 10. 失効機能付き鍵ポリシ型関数型暗号（*RKP-FE*）は，下記のアルゴリズムによって構成される．

Setup セキュリティパラメータ 1^λ と属性フォーマット $\overrightarrow{n} = (d; n_1, \ldots, n_d)$ を入力とし，公開パラメータ mpk，マスタ秘密鍵 msk，状態 st を出力．

KeyGen 公開パラメータ mpk，マスタ秘密鍵 msk，状態 st，アクセス構造 $\mathbb{S} := (M, \rho)$ を入力とし，プロキシ鍵 $\mathsf{pxk}_{\mathbb{S},KID}$ とユーザ秘密鍵 sk_{KID}，鍵 *ID:* KID，状態 st を出力．

Enc 公開パラメータ mpk，属性集合 $\Gamma := \{(t, \overrightarrow{x}_t) | \overrightarrow{x}_t \in \mathbb{F}_q^{n_t} \backslash \{\overrightarrow{0}\}, 1 \leq t \leq d\}$，メッセージ空間 msg から選んだ平文 m を入力とし，暗号文 ct_Γ を出力．

Trans 公開パラメータ mpk，暗号文 ct_Γ，プロキシ鍵 $\mathsf{pxk}_{\mathbb{S},KID}$ を入力とし，属性集合 Γ がアクセス構造 $\mathbb{S}$ を満たすなら部分復号データ ct'_{KID} を出力し，満たさないなら $\perp$ を出力．

Dec 公開パラメータ mpk，ユーザ秘密鍵 sk_{KID}，部分復号データ ct'_{KID} を入力とし，平文 m か $\perp$ を出力．

上記アルゴリズムにおいて，Correctness property は下記のように定義される．

Definition 11. すべての $(\mathsf{mpk}, \mathsf{msk}, \mathsf{st}) \xleftarrow{\mathsf{R}} \mathbf{Setup}(1^\lambda, \overrightarrow{n})$，すべてのアクセス構造 $\mathbb{S}$，すべてのプロキシ鍵とユーザ秘密鍵ペア $(\mathsf{pxk}_{\mathbb{S},KID}, \mathsf{sk}_{KID}, KID, \mathsf{st}) \xleftarrow{\mathsf{R}} \mathbf{KeyGen}(\mathsf{mpk}, \mathsf{msk}, \mathsf{st}, \mathbb{S})$，すべてのメッセージ m，アクセス構造 $\mathbb{S}$ が受理するすべての属性集合 Γ，すべての暗号文 $\mathsf{ct}_\Gamma \xleftarrow{\mathsf{R}} \mathbf{Enc}(\mathsf{mpk}, m, \Gamma)$ に対して，$m = \mathbf{Dec}(\mathsf{mpk}, \mathsf{sk}_{KID}, \mathbf{Trans}(\mathsf{mpk}, \mathsf{pxk}_{\mathbb{S},KID}, \mathsf{ct}_\Gamma))$ が成り立つ．

また，上記の失効機能付き鍵ポリシ型関数型暗号の IND-CPA 攻撃者に対する安全性は，下記のゲームによって定義される．

Definition 12. 次のゲームにおいて，多項式時間攻撃者 $\mathcal{A}$ のアドバンテージが *negligible* であるなら，その失効機能付き鍵ポリシ型関数型暗号方式は *IND-CPA* 安全であるという．

Setup 挑戦者 $\mathcal{C}$ は，***Setup*** を実行して公開パラメータ mpk，マスタ秘密鍵 msk，状態 st を生成し，公開パラメータ mpk を攻撃者 $\mathcal{A}$ に与える．

Phase1 攻撃者 $\mathcal{A}$ は，適応的に多項式回だけアクセス構造 $\mathbb{S}_\iota := (M_\iota, \rho_\iota)$ に対して以下のクエリを実施できる．

- Create($\mathbb{S}_\iota$)：挑戦者 $\mathcal{C}$ は，アクセス構造 $\mathbb{S}_\iota$ からプロキシ鍵 $\mathsf{pxk}_{\mathbb{S}_\iota,KID_\iota}$ とユーザ秘密鍵 sk_{KID_ι} を生成し，手元に保管する．
- CorruptProxyKey(ι)：挑戦者 $\mathcal{C}$ は，ι 番目に生成したプロキシ鍵 $\mathsf{pxk}_{\mathbb{S}_\iota,KID_\iota}$ を攻撃者 $\mathcal{A}$ に対して送付する．
- CorruptSecretKey(ι)：挑戦者 $\mathcal{C}$ は，ι 番目に生成したユーザ秘密鍵 sk_{KID_ι} を攻撃者 $\mathcal{A}$ に送付する．

Challenge 攻撃者 $\mathcal{A}$ は，チャレンジする平文 $m^{(0)}$, $m^{(1)}$ と属性集合 Γ^* を選び，挑戦者 $\mathcal{C}$ に送付する．挑戦者 $\mathcal{C}$ は，一様にビット $b \in \{0,1\}$ を選び，平文 $m^{(b)}$ を暗号化して，暗号文 $\mathsf{ct}_{\Gamma^*}^{(b)}$ を攻撃者 $\mathcal{A}$ に送付する．ただし，自明な攻撃を防ぐため，プロキシ鍵 $\mathsf{pxk}_{\mathbb{S}_\iota,KID_\iota}$ とユーザ秘密鍵 sk_{KID_ι} のペアを取得したアクセス構造 $\mathbb{S}_\iota$ を満たすような属性集合 Γ^* を選んではならない．

Phase2 攻撃者 $\mathcal{A}$ は，*Phase1* と同様にクエリを実行する．ただし，属性集合 Γ^* を満たすようなプロキシ鍵 $\mathsf{pxk}_{\mathbb{S}_\iota,KID_\iota}$ とユーザ秘密鍵 sk_{KID_ι} のペアをクエリしてはならない．

Guess 攻撃者 $\mathcal{A}$ は，ビット b の推測値 b' を出力する．

上記ゲームにおいて，攻撃者 $\mathcal{A}$ のアドバンテージは，セキュリティパラメータ λ に対して $\mathsf{Adv}_{\mathcal{A}}^{\mathsf{RKP-FE}}(\lambda) := |\mathsf{Pr}[b' = b] - 1/2|$ で定義される．

4.2.2 実現方式

4.1 節で述べたアイデアにより構成した失効機能付き鍵ポリシ型関数型暗号のアルゴリズムを下記に示す．

Setup$(1^\lambda, \overrightarrow{n} = (d; n_1, \ldots, n_d))$：

$(\mathsf{param}_{\overrightarrow{n}}, \{\mathbb{B}_t, \mathbb{B}_t^*\}_{t=0,\ldots,d}) \xleftarrow{\mathsf{R}} \mathcal{G}_{\mathsf{ob}}(1^\lambda, \overrightarrow{n})$,
$\hat{\mathbb{B}}_0 := (\boldsymbol{b}_{0,1}, \boldsymbol{b}_{0,3}, \boldsymbol{b}_{0,5})$,
$\hat{\mathbb{B}}_t := (\boldsymbol{b}_{t,1}, \ldots, \boldsymbol{b}_{t,n_t}, \boldsymbol{b}_{t,3n_t+1})$ for $t = 1, \ldots, d$,
$\hat{\mathbb{B}}_0^* := (\boldsymbol{b}_{0,1}^*, \boldsymbol{b}_{0,3}^*, \boldsymbol{b}_{0,4}^*)$,
$\hat{\mathbb{B}}_t^* := (\boldsymbol{b}_{t,1}^*, \ldots, \boldsymbol{b}_{t,n_t}^*, \boldsymbol{b}_{t,2n_t+1}^*, \ldots, \boldsymbol{b}_{t,3n_t}^*)$ for $t = 1, \ldots, d$,
$\mathsf{mpk} := (1^\lambda, \mathsf{param}_{\overrightarrow{n}}, \{\hat{\mathbb{B}}_t\}_{t=0,\ldots,d})$,
$\mathsf{msk} := \{\hat{\mathbb{B}}_t^*\}_{t=0,\ldots,d}$,
$\mathsf{st} := 0$,
return $\mathsf{mpk}, \mathsf{msk}, \mathsf{st}$.

KeyGen$(\mathsf{mpk}, \mathsf{msk}, \mathsf{st}, \mathbb{S})$：

$\overrightarrow{f} \xleftarrow{\mathsf{U}} \mathbb{F}_q^r$, $\overrightarrow{s}^T := (s_1, \ldots, s_\ell)^T := M \cdot \overrightarrow{f}^T$,
$s_0 := \overrightarrow{1} \cdot \overrightarrow{f}^T, \eta_0 \xleftarrow{\mathsf{U}} \mathbb{F}_q$,
$\boldsymbol{k}_0^* := (-s_0, 0, 1, \eta_0, 0)_{\mathbb{B}_0^*}$,
for $i = 1, \ldots, \ell$;
  if $\rho(i) = (t, \overrightarrow{v}_i)$,
    $\theta_i \xleftarrow{\mathsf{U}} \mathbb{F}_q$, $\overrightarrow{\eta}_i \xleftarrow{\mathsf{U}} \mathbb{F}_q^{n_t}$,
    $\boldsymbol{k}_i^* := (s_i \overrightarrow{e}_{t,1} + \theta_i \overrightarrow{v}_i, 0^{n_t}, \overrightarrow{\eta_i}, 0)_{\mathbb{B}_t^*}$,
  if $\rho(i) = \neg(t, \overrightarrow{v}_i)$,
    $\overrightarrow{\eta}_i \xleftarrow{\mathsf{U}} \mathbb{F}_q^{n_t}$,
    $\boldsymbol{k}_i^* := (s_i \overrightarrow{v}_i, 0^{n_t}, \overrightarrow{\eta_i}, 0)_{\mathbb{B}_t^*}$,
$KID := \mathsf{st}$,
$\mathsf{pxk}_{\mathbb{S},KID} := (\mathbb{S}, \boldsymbol{k}_1^*, \ldots, \boldsymbol{k}_\ell^*)$,
$\mathsf{sk}_{KID} := (\boldsymbol{k}_0^*)$,
$\mathsf{st} := \mathsf{st} + 1$,
return $\mathsf{pxk}_{\mathbb{S},KID}, \mathsf{sk}_{KID}, KID, \mathsf{st}$.

Enc$(\mathsf{mpk}, m, \Gamma)$：

$\omega, \varphi_0, \varphi_t, \zeta \xleftarrow{\mathsf{U}} \mathbb{F}_q$ for $(t, \overrightarrow{x}_t) \in \Gamma$,
$\boldsymbol{c}_0 := (\omega, 0, \zeta, 0, \varphi_0)_{\mathbb{B}_0}$,
$\boldsymbol{c}_t := (\omega \overrightarrow{x}_t, 0^{n_t}, 0^{n_t}, \varphi_t)_{\mathbb{B}_t}$ for $(t, \overrightarrow{x}_t) \in \Gamma$,
$c_{d+1} := g_T^\zeta m$,
return $\mathsf{ct}_\Gamma := (\Gamma, \boldsymbol{c}_0, \{\boldsymbol{c}_t\}_{(t,\overrightarrow{x}_t)\in\Gamma}, c_{d+1})$.

Trans$(\mathsf{mpk}, \mathsf{pxk}_{\mathbb{S},KID}, \mathsf{ct}_\Gamma)$：

もし $\mathbb{S}$ が Γ を受け入れるなら，下記に示すような I と $\{\alpha_i\}_{i\in I}$ を計算する：
$\overrightarrow{1} = \sum_{i\in I} \alpha_i M_i$,
$I \subseteq \{i \in \{1, \ldots, \ell\} | [\rho(i) = (t, \overrightarrow{v}_i) \wedge (t, \overrightarrow{x}_t) \in \Gamma \wedge \overrightarrow{v}_i \cdot \overrightarrow{x}_t = 0] \vee [\rho(i) = \neg(t, \overrightarrow{v}_i) \wedge (t, \overrightarrow{x}_t) \in \Gamma \wedge \overrightarrow{v}_i \cdot \overrightarrow{x}_t \neq 0]\}$,
$K_2 := \prod_{i\in I\wedge\rho(i)=(t,\overrightarrow{v}_i)} e(\boldsymbol{c}_t, \boldsymbol{k}_i^*)^{\alpha_i} \cdot \prod_{i\in I\wedge\rho(i)=\neg(t,\overrightarrow{v}_i)} e(\boldsymbol{c}_t, \boldsymbol{k}_i^*)^{\alpha_i/(\overrightarrow{v}_i\cdot\overrightarrow{x}_t)}$,
return $\mathsf{ct}'_{KID} := (\boldsymbol{c}_0, c_{d+1}, K_2)$.

Dec$(\mathsf{mpk}, \mathsf{sk}_{KID}, \mathsf{ct}'_{KID})$：

$K_1 := e(\boldsymbol{c}_0, \boldsymbol{k}_0^*)$,
$K = K_1 \cdot K_2$,
return $m' = c_{d+1}/K$.

なお，上記アルゴリズムにおいて，ベクトル $\overrightarrow{x}_t := (x_{t,1}, \ldots, x_{t,n_t})$ は，$x_{t,1} = 1$ になるように正規化されていると仮定する．

［**Correctness property**］

上記アルゴリズムにおいて，属性集合 Γ がアクセス構造 $\mathbb{S}$ を満たすなら，下記が成り立つ．

$$K_1 = e(\boldsymbol{c}_0, \boldsymbol{k}_0^*) = g_T^{-\omega s_0 + \zeta}$$

$$K_2 = \prod_{i\in I\wedge\rho(i)=(t,\overrightarrow{v}_i)} e(\boldsymbol{c}_t, \boldsymbol{k}_i^*)^{\alpha_i} \cdot \prod_{i\in I\wedge\rho(i)=\neg(t,\overrightarrow{v}_i)} e(\boldsymbol{c}_t, \boldsymbol{k}_i^*)^{\alpha_i/(\overrightarrow{v}_i\cdot\overrightarrow{x}_t)}$$

$$= \prod_{i\in I\wedge\rho(i)=(t,\overrightarrow{v}_i)} g_T^{\omega\alpha_i(s_i+\theta_i\overrightarrow{v}_i\cdot\overrightarrow{x}_t)} \cdot \prod_{i\in I\wedge\rho(i)=\neg(t,\overrightarrow{v}_i)} g_T^{\omega\alpha_i s_i(\overrightarrow{v}_i\cdot\overrightarrow{x}_t)/(\overrightarrow{v}_i\cdot\overrightarrow{x}_t)}$$
$$= g_T^{\omega(\sum_{i\in I}\alpha_i s_i)} = g_T^{\omega s_0}$$
$$K = K_1\cdot K_2 = g_T^{-\omega s_0+\zeta}\cdot g_T^{\omega s_0} = g_T^{\zeta}$$

それゆえ，

$$c_{d+1}/K = g_T^{\zeta} m / g_T^{\zeta} = m$$

よって，Correctness property が成り立つ．

4.2.3 安全性

上記提案方式の安全性は，下記のように示すことができる．

Theorem 1. 失効機能付き鍵ポリシ型関数型暗号に対して選択平文攻撃を行う確率的多項式時間攻撃者 $\mathcal{A}$ を仮定したとき，攻撃者 $\mathcal{A}$ のアドバンテージ $\mathsf{Adv}_{\mathcal{A}}^{\mathsf{RKP-FE}}(\lambda)$ は下記に示すように *DLIN* 仮定のもとで *negligible* であり，提案方式は *IND-CPA* 安全である．

$$\mathsf{Adv}_{\mathcal{A}}^{\mathsf{RKP-FE}}(\lambda) \le \mathsf{Adv}_{\mathcal{E}_1}^{\mathsf{DLIN}}(\lambda) + \sum_{h=0}^{\nu-1}\left(\mathsf{Adv}_{\mathcal{E}_{2,h}^{+}}^{\mathsf{DLIN}}(\lambda)+\mathsf{Adv}_{\mathcal{E}_{2,h+1}}^{\mathsf{DLIN}}(\lambda)\right)+\epsilon$$

ここで，ν は攻撃者 $\mathcal{A}$ の鍵クエリの回数であり，$\epsilon = (2d\nu+16\nu+d+7)/q$ である．

本提案方式は，Takashima ら [18] が提案した関数型暗号の復号鍵に対してのみ修正を行っているので，関数型暗号の証明をベースとして安全性証明を実施する．本証明は Dual System Encryption に基づいており，Game0 が Definition 12 に示したオリジナルのゲームであり，チャレンジ暗号文を semi-functional に変化させた Game1 に変化させる．その後，プロキシ鍵もしくはユーザ秘密鍵を 1 個ずつ semi-functional に変えていくが，最初に 1 回目にクエリされたプロキシ鍵もしくはユーザ秘密鍵を pre-semi-functional に変えた Game2-0$^+$，次に semi-functional に変えた Game2-1 に変化させる．これをすべてのクエリに対して実施し，Game2-ν まで変化させる．最後に，チャレンジ暗号文を random に変化させた Game3 まで到達する．

ここで課題となるのは，オリジナルの関数型暗号の証明では，Game2-h^+ においてチャレンジ暗号文を復号できるアクセス構造を指定して復号鍵をクエリしたときにのみ，チャレンジ暗号文の要素 $\boldsymbol{c}_0$ の第 2 成分 r_0，復号鍵の要素 $\boldsymbol{k}_0^*$ の第 2 成分 w_0，同じく復号鍵の要素 $\boldsymbol{k}_i^*$ の第 (n_t+1)〜$2n_t$ 成分に含まれる $\{a_i\}_{i=1,\cdots,\ell}$ から導出される要素 a_0 の 3 点が，$w_0 = a_0/r_0$ という correlation を起こしてしまい，正しく semi-functional key がシミュレートできない点である．言い換えれば，3 個の変数のうち 2 個が独立変数で，1 個が従属変数になるという correlation が存在する．我々の実現方式では，これらの correlation を起こす要素をチャレンジ暗号文，プロキシ鍵，ユーザ秘密鍵にそれぞれ 1 個ずつ分散配置するアルゴリズム構成としたことで，チャレンジ暗号文を復号できるアクセス構造を指定した場合においても，自明な攻撃を防ぐために攻撃者はプロキシ鍵かユーザ秘密鍵のいずれか一方しかクエリできないという制約から，上記で述べた correlation を起こす 3 変数のうち 2 変数しか入手することができず，攻撃者のビューからは correlation が見えなくなる．そのため，Dual system encryption の枠組みを用いて正しくゲームがシミュレートでき，安全性が証明できる．

Proof. Theorem 1 を証明するためのゲーム列を**図 5** に示す．Game0 が Definition 12 に示したオリジナルのゲームであり，この要素を徐々に変えていき，Game3 まで変化させる．前のゲームから変化する点は，四角枠で囲った部分である．Game0，1，2-h，2-h^+，3 の攻撃者 $\mathcal{A}$ のアドバンテージを $\mathsf{Adv}_{\mathcal{A}}^{(0)}(\lambda), \mathsf{Adv}_{\mathcal{A}}^{(1)}(\lambda), \mathsf{Adv}_{\mathcal{A}}^{(2-h)}(\lambda), \mathsf{Adv}_{\mathcal{A}}^{(2-h^+)}(\lambda), \mathsf{Adv}_{\mathcal{A}}^{(3)}(\lambda)$ とする．このとき，それぞれのアドバンテージの違いは後で示す Lemma 3，4，6，7，8 のようになるため，$\mathsf{Adv}_{\mathcal{A}}^{\mathsf{RKP-FE}}(\lambda)$ は下記の計算できる．

$$\begin{aligned}
\mathsf{Adv}_{\mathcal{A}}^{\mathsf{RKP-FE}}(\lambda) &= \mathsf{Adv}_{\mathcal{A}}^{(0)}(\lambda)\\
&\le \left|\mathsf{Adv}_{\mathcal{A}}^{(0)}(\lambda)-\mathsf{Adv}_{\mathcal{A}}^{(1)}(\lambda)\right|\\
&\quad+\sum_{h=0}^{\nu-1}\left|\mathsf{Adv}_{\mathcal{A}}^{(2-h)}(\lambda)-\mathsf{Adv}_{\mathcal{A}}^{(2-h^+)}(\lambda)\right|\\
&\quad+\sum_{h=0}^{\nu-1}\left|\mathsf{Adv}_{\mathcal{A}}^{(2-h^+)}(\lambda)-\mathsf{Adv}_{\mathcal{A}}^{(2-(h+1))}(\lambda)\right|\\
&\quad+\left|\mathsf{Adv}_{\mathcal{A}}^{(2-\nu)}(\lambda)-\mathsf{Adv}_{\mathcal{A}}^{(3)}(\lambda)\right|\\
&\quad+\left|\mathsf{Adv}_{\mathcal{A}}^{(3)}(\lambda)\right|\\
&\le \mathsf{Adv}_{\mathcal{B}_1}^{\mathsf{P1}}(\lambda)+\sum_{h=0}^{\nu-1}\mathsf{Adv}_{\mathcal{B}_{2,h}^{+}}^{\mathsf{P2}}(\lambda)\\
&\quad+\sum_{h=0}^{\nu-1}\mathsf{Adv}_{\mathcal{B}_{2,h+1}}^{\mathsf{P2}}(\lambda)+(2d\nu+6\nu+1)/q\\
&\le \mathsf{Adv}_{\mathcal{E}_1}^{\mathsf{DLIN}}(\lambda)\\
&\quad+\sum_{h=0}^{\nu-1}\left(\mathsf{Adv}_{\mathcal{E}_{2,h}^{+}}^{\mathsf{DLIN}}(\lambda)+\mathsf{Adv}_{\mathcal{E}_{2,h+1}}^{\mathsf{DLIN}}(\lambda)\right)\\
&\quad+(2d\nu+16\nu+d+7)/q
\end{aligned}$$

以上により，Theorem 1 が成り立つ． □

本定理で利用した Lemma は，付録 A.1 に示す．

4.3 失効機能付き暗号文ポリシ型関数型暗号

暗号文ポリシ型関数型暗号に対して失効機能を付加したアルゴリズム，およびその安全性について示す．

4.3.1 アルゴリズム定義とセキュリティ定義

失効機能付き暗号文ポリシ型関数型暗号（RCP-FE）の

Game0: オリジナルのゲーム．チャレンジ暗号文は，正規の手順で生成した基底を用いて，下記のように正規の暗号文を生成する．

$\boldsymbol{c}_0 := (\omega, 0, \zeta, 0, \varphi_0)_{\mathbb{B}_0}$,

$\boldsymbol{c}_t := (\omega \overrightarrow{x}_t, 0^{n_t}, 0^{n_t}, \varphi_t)_{\mathbb{B}_t}$ for $(t, \overrightarrow{x}_t) \in \Gamma$,

$c_{d+1} := g_T^{\zeta} m^{(b)}$,

同様に，プロキシー鍵とユーザ秘密鍵も正規の手順で生成した基底を用いて正規の手順で生成する．

$\boldsymbol{k}_0^* := (-s_0, 0, 1, \eta_0, 0)_{\mathbb{B}_0^*}$,

for $i = 1, \ldots, \ell$;

if $\rho(i) = (t, \overrightarrow{v}_i)$, $\boldsymbol{k}_i^* := (s_i \overrightarrow{e}_{t,1} + \theta_i \overrightarrow{v}_i, 0^{n_t}, \overrightarrow{\eta_i}, 0)_{\mathbb{B}_t^*}$,

if $\rho(i) = \neg(t, \overrightarrow{v}_i)$, $\boldsymbol{k}_i^* := (s_i \overrightarrow{v}_i, 0^{n_t}, \overrightarrow{\eta_i}, 0)_{\mathbb{B}_t^*}$,

$\mathsf{pxk}_{\mathbb{S},KID} := (\mathbb{S}, \boldsymbol{k}_1^*, \ldots, \boldsymbol{k}_\ell^*)$,

$\mathsf{sk}_{KID} := (\boldsymbol{k}_0^*)$,

Game1: チャレンジ暗号文を下記の様に生成する点だけが Game0 と異なる．

$r_0 \xleftarrow{\cup} \mathbb{F}_q$, $\overrightarrow{r}_t \xleftarrow{\cup} \mathbb{F}_q^{n_t}$,

$\boldsymbol{c}_0 := (\omega, \boxed{r_0}, \zeta, 0, \varphi_0)_{\mathbb{B}_0}$,

$\boldsymbol{c}_t := (\omega \overrightarrow{x}_t, \boxed{\overrightarrow{r}_t}, 0^{n_t}, \varphi_t)_{\mathbb{B}_t}$ for $(t, \overrightarrow{x}_t) \in \Gamma$,

Game2-h^+ $(h = 0, \cdots, \nu - 1)$: プロキシー鍵，ユーザ秘密鍵，チャレンジ暗号文の下記の要素の生成方法だけが Game2-h と異なる．なお，Game2-0 は Game1 と同一である．

$w_0 \xleftarrow{\cup} \mathbb{F}_q$, $\overrightarrow{g} \xleftarrow{\cup} \mathbb{F}_q^r$, $\overrightarrow{a}^T := (a_1, \ldots, a_\ell)^T := M \cdot \overrightarrow{g}^T, \pi_i \xleftarrow{\cup} \mathbb{F}_q (i = 1, \ldots, \ell)$,

$Z_t \xleftarrow{\cup} GL(n_t, \mathbb{F}_q), U_t := (Z_t^{-1})^T$ for $t = 1, \ldots, d$,

$\boldsymbol{k}_0^* := (-s_0, \boxed{w_0}, 1, \eta_0, 0)_{\mathbb{B}_0^*}$,

for $i = 1, \ldots, \ell$;

if $\rho(i) = (t, \overrightarrow{v}_i)$, $\boldsymbol{k}_i^* := (s_i \overrightarrow{e}_{t,1} + \theta_i \overrightarrow{v}_i, \boxed{(a_i \overrightarrow{e}_{t,1} + \pi_i \overrightarrow{v}_i) \cdot Z_t}, \overrightarrow{\eta_i}, 0)_{\mathbb{B}_t^*}$,

if $\rho(i) = \neg(t, \overrightarrow{v}_i)$, $\boldsymbol{k}_i^* := (s_i \overrightarrow{v}_i, \boxed{a_i \overrightarrow{v}_i \cdot Z_t}, \overrightarrow{\eta_i}, 0)_{\mathbb{B}_t^*}$,

$\boldsymbol{c}_t := (\omega \overrightarrow{x}_t, \boxed{\overrightarrow{x}_t \cdot U_t}, 0^{n_t}, \varphi_t)_{\mathbb{B}_t}$ for $(t, \overrightarrow{x}_t) \in \Gamma$,

Game2-$(h+1)$ $(h = 0, \cdots, \nu - 1)$: プロキシー鍵とチャレンジ暗号文の下記の要素の生成方法だけが Game2-h^+ と異なる．

$\overrightarrow{r}_t \xleftarrow{\cup} \mathbb{F}_q^{n_t}$,

$\boldsymbol{k}_0^* := (-s_0, w_0, 1, \eta_0, 0)_{\mathbb{B}_0^*}$,

for $i = 1, \ldots, \ell$;

if $\rho(i) = (t, \overrightarrow{v}_i)$, $\boldsymbol{k}_i^* := (s_i \overrightarrow{e}_{t,1} + \theta_i \overrightarrow{v}_i, \boxed{0^{n_t}}, \overrightarrow{\eta_i}, 0)_{\mathbb{B}_t^*}$,

if $\rho(i) = \neg(t, \overrightarrow{v}_i)$, $\boldsymbol{k}_i^* := (s_i \overrightarrow{v}_i, \boxed{0^{n_t}}, \overrightarrow{\eta_i}, 0)_{\mathbb{B}_t^*}$,

$\boldsymbol{c}_t := (\omega \overrightarrow{x}_t, \boxed{\overrightarrow{r}_t}, 0^{n_t}, \varphi_t)_{\mathbb{B}_t}$ for $(t, \overrightarrow{x}_t) \in \Gamma$,,

Game3: チャレンジ暗号文の下記の要素の生成方法だけが Game2-ν と異なる．

$\zeta' \xleftarrow{\cup} \mathbb{F}_q$,

$\boldsymbol{c}_0 := (\omega, r_0, \boxed{\zeta'}, 0, \varphi_0)_{\mathbb{B}_0}$,

$c_{d+1} := g_T^{\zeta} m^{(b)}$,

図 5 失効機能対応鍵ポリシ型関数型暗号の安全性証明のゲーム列

Fig. 5 Sequence of games for the revocable key-policy functional encryption.

アルゴリズム定義を下記に示す．

Definition 13. 失効機能付き暗号文ポリシ型関数型暗号（*RCP-FE*）は，下記のアルゴリズムによって構成される．

Setup セキュリティパラメータ 1^λ と属性フォーマット $\overrightarrow{n} = (d; n_1, \ldots, n_d)$ を入力とし，公開パラメータ mpk とマスタ秘密鍵 msk，状態 st を出力．

KeyGen 公開パラメータ mpk，マスタ秘密鍵 msk，状態 st，属性集合 $\Gamma := \{(t, \overrightarrow{x}_t) | \overrightarrow{x}_t \in \mathbb{F}_q^{n_t}, 1 \le t \le d\}$ を入力とし，プロキシ鍵 $\mathsf{pxk}_{\Gamma,KID}$，ユーザ秘密鍵 sk_{KID}，鍵 *ID:* KID，状態 st を出力．

Enc 公開パラメータ mpk，アクセス構造 $\mathbb{S} := (M, \rho)$，メッセージ空間 msg から選んだ平文 m を入力とし，暗号文 $\mathsf{ct}_{\mathbb{S}}$ を出力．

Trans 公開パラメータ mpk，暗号文 $\mathsf{ct}_{\mathbb{S}}$，プロキシ鍵 $\mathsf{pxk}_{\Gamma,KID}$ を入力とし，属性集合 Γ がアクセス構造 $\mathbb{S}$ を満たすなら部分復号データ ct'_{KID} を出力し，満たさないなら $\perp$ を出力．

Dec 公開パラメータ mpk，ユーザ秘密鍵 sk_{KID}，部分復号データ ct'_{KID} を入力とし，平文 m か $\perp$ を出力．

上記アルゴリズムにおいて，Correctness property は下記のように定義される．

Definition 14. すべての $(\mathsf{mpk}, \mathsf{msk}, \mathsf{st}) \xleftarrow{R} \mathbf{Setup}(1^\lambda, \overrightarrow{n})$，すべての属性集合 Γ，すべてのプロキシ鍵と秘密鍵ペア $(\mathsf{pxk}_{\Gamma,KID}, \mathsf{sk}_{KID}, KID, \mathsf{st}) \xleftarrow{R} \mathbf{KeyGen}(\mathsf{mpk}, \mathsf{msk}, \mathsf{st}, \Gamma)$，すべてのメッセージ m，属性集合 Γ を受理するすべてのアクセス構造 $\mathbb{S}$，すべての暗号文 $\mathsf{ct}_{\mathbb{S}} \xleftarrow{R} \mathbf{Enc}(\mathsf{mpk}, m, \mathbb{S})$ に対して，$m = \mathbf{Dec}(\mathsf{mpk}, \mathsf{sk}_{KID}, \mathbf{Trans}(\mathsf{mpk}, \mathsf{pxk}_{\Gamma,KID}, \mathsf{ct}_{\mathbb{S}}))$ が成り立つ．

また，上記の失効機能付き暗号文ポリシ型関数型暗号の IND-CPA 攻撃者に対する安全性は，下記のゲームによって定義される．

Definition 15. 次のゲームにおいて，多項式時間攻撃者

$\mathcal{A}$ のアドバンテージが *negligible* であるなら，その失効機能付き暗号文ポリシ型関数型暗号方式は *IND-CPA* 安全であるという．

Setup 挑戦者 $\mathcal{C}$ は，***Setup*** を実行して公開パラメータ mpk，マスタ秘密鍵 msk，状態 st を生成し，公開パラメータ mpk を攻撃者 $\mathcal{A}$ に与える．

Phase1 攻撃者 $\mathcal{A}$ は，適応的に多項式回だけ属性集合 Γ に対応する以下のクエリを実施できる．

- Create(Γ_ι)：挑戦者 $\mathcal{C}$ は，属性集合 Γ_ι に対応するプロキシ鍵 $\mathsf{pxk}_{\Gamma_\iota,KID_\iota}$ とユーザ秘密鍵 sk_{KID_ι} を生成し，手元に保管する．
- CorruptProxyKey(ι)：挑戦者 $\mathcal{C}$ は，ι 番目に生成したプロキシ鍵 $\mathsf{pxk}_{\Gamma_\iota,KID_\iota}$ を攻撃者 $\mathcal{A}$ に対して送付する．
- CorruptSecretKey(ι)：挑戦者 $\mathcal{C}$ は，ι 番目に生成したユーザ秘密鍵 sk_{KID_ι} を攻撃者 $\mathcal{A}$ に送付する．

Challenge 攻撃者 $\mathcal{A}$ は，チャレンジする平文 $m^{(0)}$，$m^{(1)}$ とアクセス構造 $\mathbb{S}^*$ を選び，挑戦者 $\mathcal{C}$ に送付する．挑戦者 $\mathcal{C}$ は，一様にビット $b\in\{0,1\}$ を選び，平文 $m^{(b)}$ を暗号化して，暗号文 $\mathsf{ct}_{\mathbb{S}^*}^{(b)}$ を攻撃者 $\mathcal{A}$ に送付する．ただし，自明な攻撃を防ぐため，プロキシ鍵 $\mathsf{pxk}_{\Gamma_\iota,KID_\iota}$ とユーザ秘密鍵 sk_{KID_ι} のペアを取得した属性集合 Γ_ι が満たすようなアクセス構造 $\mathbb{S}^*$ を選んではならない．

Phase2 攻撃者 $\mathcal{A}$ は，*Phase1* と同様にクエリを実行する．ただし，アクセス構造 $\mathbb{S}^*$ を満たすような属性集合 Γ_ι に対して，プロキシ鍵 $\mathsf{pxk}_{\Gamma_\iota,KID_\iota}$ とユーザ秘密鍵 sk_{KID_ι} のペアをクエリしてはならない．

Guess 攻撃者 $\mathcal{A}$ は，ビット b の推測値 b' を出力する．

上記ゲームにおいて，攻撃者 $\mathcal{A}$ のアドバンテージは，セキュリティパラメータ λ に対して $\mathsf{Adv}_{\mathcal{A}}^{\mathsf{RCP-FE}}(\lambda):=|\mathsf{Pr}[b'=b]-1/2|$ で定義される．

4.3.2 実現方式

4.1 節で述べたアイデアにより構成した失効機能付き暗号文ポリシ型関数型暗号のアルゴリズムを下記に示す．

Setup$(1^\lambda,\overrightarrow{n}=(d;n_1,\ldots,n_d))$：
$(\mathsf{param}_{\overrightarrow{n}},\{\mathbb{B}_t,\mathbb{B}_t^*\}_{t=0,\ldots,d})\xleftarrow{\mathsf{R}}\mathcal{G}_{\mathsf{ob}}(1^\lambda,\overrightarrow{n})$,
$\hat{\mathbb{B}}_0:=(\boldsymbol{b}_{0,1},\boldsymbol{b}_{0,3},\boldsymbol{b}_{0,5})$,
$\hat{\mathbb{B}}_t:=(\boldsymbol{b}_{t,1},\ldots,\boldsymbol{b}_{t,n_t},\boldsymbol{b}_{t,3n_t+1})$ for $t=1,\ldots,d$,
$\hat{\mathbb{B}}_0^*:=(\boldsymbol{b}_{0,1}^*,\boldsymbol{b}_{0,3}^*,\boldsymbol{b}_{0,4}^*)$,
$\hat{\mathbb{B}}_t^*:=(\boldsymbol{b}_{t,1}^*,\ldots,\boldsymbol{b}_{t,n_t}^*,\boldsymbol{b}_{t,2n_t+1}^*,\ldots,\boldsymbol{b}_{t,3n_t}^*)$ for $t=1,\ldots,d$,
$\mathsf{mpk}:=(1^\lambda,\mathsf{param}_{\overrightarrow{n}},\{\hat{\mathbb{B}}_t\}_{t=0,\ldots,d})$,
$\mathsf{msk}:=\{\hat{\mathbb{B}}_t^*\}_{t=0,\ldots,d}$,
$\mathsf{st}:=0$,
return $\mathsf{mpk},\mathsf{msk},\mathsf{st}$.

KeyGen$(\mathsf{mpk},\mathsf{msk},\mathsf{st},\Gamma)$：
$\delta,\varphi_0\xleftarrow{\mathsf{U}}\mathbb{F}_q$, $\overrightarrow{\varphi}_t\xleftarrow{\mathsf{U}}\mathbb{F}_q^{n_t}$ for $(t,\overrightarrow{x}_t)\in\Gamma$,
$\boldsymbol{k}_0^*:=(\delta,0,1,\varphi_0,0)_{\mathbb{B}_0^*}$,
$\boldsymbol{k}_t^*:=(\delta\overrightarrow{x}_t,0^{n_t},\overrightarrow{\varphi}_t,0)_{\mathbb{B}_t^*}$ for $(t,\overrightarrow{x}_t)\in\Gamma$,
$KID:=\mathsf{st}$,
$\mathsf{pxk}_{\Gamma,KID}:=(\Gamma,\{\boldsymbol{k}_t^*\}_{(t,\overrightarrow{x}_t)\in\Gamma})$,
$\mathsf{sk}_{KID}:=(\boldsymbol{k}_0^*)$,
$\mathsf{st}:=\mathsf{st}+1$,
return $\mathsf{pxk}_{\Gamma,KID},\mathsf{sk}_{KID},KID,\mathsf{st}$.

Enc$(\mathsf{mpk},m,\mathbb{S})$：
$\overrightarrow{f}\xleftarrow{\mathsf{U}}\mathbb{F}_q^r$, $\overrightarrow{s}^T:=(s_1,\ldots,s_\ell)^T:=M\cdot\overrightarrow{f}^T$,
$s_0:=\overrightarrow{1}\cdot\overrightarrow{f}^T,\eta_0,\eta_i,\theta_i,\zeta\xleftarrow{\mathsf{U}}\mathbb{F}_q(i=1,\ldots,\ell)$,
$\boldsymbol{c}_0:=(-s_0,0,\zeta,0,\eta_0)_{\mathbb{B}_0}$,
for $i=1,\ldots,\ell$;
　if $\rho(i)=(t,\overrightarrow{v}_i)$,
　　$\boldsymbol{c}_i:=(s_i\overrightarrow{e}_{t,1}+\theta_i\overrightarrow{v}_i,0^{n_t},0^{n_t},\eta_i)_{\mathbb{B}_t}$,
　if $\rho(i)=\neg(t,\overrightarrow{v}_i)$,
　　$\boldsymbol{c}_i:=(s_i\overrightarrow{v}_i,0^{n_t},0^{n_t},\eta_i)_{\mathbb{B}_t}$,
$c_{d+1}:=g_T^\zeta m$,
return $\mathsf{ct}_{\mathbb{S}}:=(\mathbb{S},\boldsymbol{c}_0,\boldsymbol{c}_1,\ldots,\boldsymbol{c}_\ell,c_{d+1})$.

Trans$(\mathsf{mpk},\mathsf{pxk}_{\Gamma,KID},\mathsf{ct}_{\mathbb{S}})$：
もし $\mathbb{S}$ が Γ を受け入れるなら，下記に示すような I と $\{\alpha_i\}_{i\in I}$ を計算する：
$\overrightarrow{1}=\sum_{i\in I}\alpha_iM_i$,
$I\subseteq\{i\in\{1,\ldots,\ell\}|[\rho(i)=(t,\overrightarrow{v}_i)\wedge(t,\overrightarrow{x}_t)\in\Gamma\wedge\overrightarrow{v}_i\cdot\overrightarrow{x}_t=0]\vee[\rho(i)=\neg(t,\overrightarrow{v}_i)\wedge(t,\overrightarrow{x}_t)\in\Gamma\wedge\overrightarrow{v}_i\cdot\overrightarrow{x}_t\neq0]\}$,
$K_2=\prod_{i\in I\wedge\rho(i)=(t,\overrightarrow{v}_i)}e(\boldsymbol{c}_i,\boldsymbol{k}_t^*)^{\alpha_i}\cdot\prod_{i\in I\wedge\rho(i)=\neg(t,\overrightarrow{v}_i)}e(\boldsymbol{c}_i,\boldsymbol{k}_t^*)^{\alpha_i/(\overrightarrow{v}_i\cdot\overrightarrow{x}_t)}$,
return $\mathsf{ct}'_{KID}:=(\boldsymbol{c}_0,c_{d+1},K_2)$.

Dec$(\mathsf{mpk},\mathsf{sk}_{KID},\mathsf{ct}'_{KID})$：
$K_1:=e(\boldsymbol{c}_0,\boldsymbol{k}_0^*)$,
$K=K_1\cdot K_2$,
return $m'=c_{d+1}/K$.

なお，上記アルゴリズムにおいて，ベクトル $\overrightarrow{x}_t:=(x_{t,1},\ldots,x_{t,n_t})$ は，$x_{t,1}=1$ になるように正規化されていると仮定する．また，ベクトル $\overrightarrow{v}_i:=(v_{i,1},\ldots,v_{i,n_t})$ は，$v_{i,n_t}\neq0$ であると仮定する．

［**Correctness property**］

上記アルゴリズムにおいて，属性集合 Γ がアクセス構造 $\mathbb{S}$ を満たすなら，下記が成り立つ．

$$
\begin{aligned}
K_1 &= e(\boldsymbol{c}_0,\boldsymbol{k}_0^*)=g_T^{-\delta s_0+\zeta}\\
K_2 &= \prod_{i\in I\wedge\rho(i)=(t,\overrightarrow{v}_i)}e(\boldsymbol{c}_i,\boldsymbol{k}_t^*)^{\alpha_i}\\
&\quad\cdot\prod_{i\in I\wedge\rho(i)=\neg(t,\overrightarrow{v}_i)}e(\boldsymbol{c}_i,\boldsymbol{k}_t^*)^{\alpha_i/(\overrightarrow{v}_i\cdot\overrightarrow{x}_t)}\\
&= \prod_{i\in I\wedge\rho(i)=(t,\overrightarrow{v}_i)}g_T^{\delta\alpha_i(s_i+\theta_i\overrightarrow{v}_i\cdot\overrightarrow{x}_t)}\\
&\quad\cdot\prod_{i\in I\wedge\rho(i)=\neg(t,\overrightarrow{v}_i)}g_T^{\delta\alpha_is_i(\overrightarrow{v}_i\cdot\overrightarrow{x}_t)/(\overrightarrow{v}_i\cdot\overrightarrow{x}_t)}
\end{aligned}
$$

$$= g_T^{\delta \sum_{i \in I} \alpha_i s_i} = g_T^{\delta s_0}$$

$$K = K_1 \cdot K_2 = g_T^{-\delta s_0 + \zeta} \cdot g_T^{\delta s_0} = g_T^{\zeta}$$

それゆえ，

$$c_{d+1}/K = g_T^{\zeta} m / g_T^{\zeta} = m$$

よって，Correctness property が成り立つ．

4.3.3 安全性

上記提案方式の安全性は，下記のように示すことができる．

Theorem 2. 失効機能付き暗号文ポリシ型関数型暗号に対して選択平文攻撃を行う確率的多項式時間攻撃者 $\mathcal{A}$ を仮定したとき，攻撃者 $\mathcal{A}$ のアドバンテージ $\mathsf{Adv}_{\mathcal{A}}^{\mathsf{RCP-FE}}(\lambda)$ は下記に示すように *DLIN* 仮定のもとで *negligible* であり，提案方式は *IND-CPA* 安全である．

$$\mathsf{Adv}_{\mathcal{A}}^{\mathsf{RCP-FE}}(\lambda) \leq \mathsf{Adv}_{\mathcal{E}_1}^{\mathsf{DLIN}}(\lambda) + \sum_{h=0}^{\nu-1} \left(\mathsf{Adv}_{\mathcal{E}_{2,h}^{+}}^{\mathsf{DLIN}}(\lambda) + \mathsf{Adv}_{\mathcal{E}_{2,h+1}}^{\mathsf{DLIN}}(\lambda) \right) + \epsilon$$

ここで，ν は攻撃者 $\mathcal{A}$ の鍵クエリの回数であり，$\epsilon = (2d\nu + 16\nu + 2d + 8)/q$ である．

本定理の証明は，付録 A.2 に示す．

5. 鍵管理方式

システム構成がシンプルで分かりやすい失効機能付き暗号文ポリシ型関数型暗号を例に，提案方式を実システムに適用する場合の鍵管理方式について述べる．基本的な流れは鍵ポリシ型関数型暗号でも同一である．

5.1 システムセットアップ手順

システムのセットアップとして，提案方式の鍵生成を行う手順を**図 6** に示す．初めに，ユーザの属性情報が管理されているディレクトリサーバから，どの属性情報を用いるかを抽出し，属性番号表を作成する．この属性番号表では，属性情報と，属性値をベクトル表現 $(t, \overrightarrow{x}_t)$ する際の属性番号 t との対応付けを行う．また，本提案方式では属性値の一致をベクトル内積値で判定する必要があるため，$\overrightarrow{x}_t = (1, '\text{属性値}')$，$\overrightarrow{v}_t = ('\text{属性値}', -1)$ とベクトル化するルールとし，$n_t = 2$ $(1 \leq t \leq d)$ とする．

この属性番号表より，セキュリティパラメータ 1^λ と属性フォーマット $\overrightarrow{n}$ を決定し，**Setup** アルゴリズムを実行して，公開パラメータ mpk とマスタ秘密鍵 msk と状態 st を生成する．このすべてのパラメータをストレージで安全に保管するとともに，公開パラメータと属性番号表は暗号化・復号処理に必要なためユーザに対して公開する．

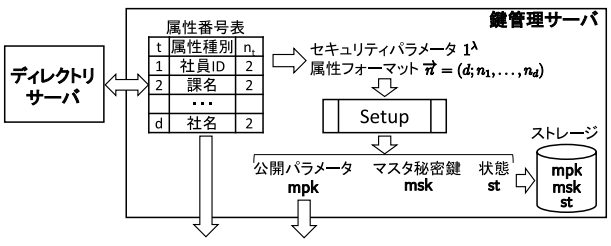


図 6 鍵生成などシステムセットアップの流れ
Fig. 6 The flow for system setup and key generation.

5.2 ユーザ追加手順

ユーザに対して新たにユーザ秘密鍵を発行する手順を**図 7** に示す．鍵管理サーバは，システム管理者から社員 ID を受領したら，ディレクトリサービスから該当ユーザの属性を取得する．本例では，社員 ID “$ID001$” のユーザが指定されたと仮定する．そして，セットアップ時に定めた属性番号表を参照して属性集合 Γ_{ID001} を生成したのちに，ストレージに保管された公開パラメータ mpk とマスタ秘密鍵 msk と状態 st を取り出して **KeyGen** アルゴリズムを実行し，プロキシ鍵 $\mathsf{pxk}_{\Gamma_{ID001}, KID:=1}$ とユーザ秘密鍵 $\mathsf{sk}_{KID:=1}$ と鍵 ID: $KID := 1$ と更新された状態 st を生成する．プロキシ鍵 $\mathsf{pxk}_{\Gamma_{ID001}, KID:=1}$ はプロキシサーバへ登録を要求し，ユーザ秘密鍵 $\mathsf{sk}_{KID:=1}$ はユーザへ配布する．

プロキシ鍵 $\mathsf{pxk}_{\Gamma_{ID001}, KID:=1}$ を受け取ったプロキシサーバは，自身が管理するプロキシ鍵リストに鍵 ID・社員 ID と関連付けて保管を行う．また，ユーザは，受領したユーザ秘密鍵 $\mathsf{sk}_{KID:=1}$ を鍵 ID と関連付けて自身が管理するストレージに安全に保管する．

5.3 暗号化データ復号手順

ユーザが暗号化データにアクセスする際の手順を**図 8** に示す．初めに社員 ID が “$ID001$” であるユーザは，自身が復号権限を持つ暗号化データ $\mathsf{ct}_{\mathbb{S}}$ を指定して，プロキシサーバに取得を依頼する．要求を受けたプロキシサーバは，クラウドストレージから暗号化データ $\mathsf{ct}_{\mathbb{S}}$ をダウンロードし，管理するプロキシ鍵リストから社員 ID=“$ID001$”，かつ属性集合 Γ_{ID001} が暗号化データのアクセス構造 $\mathbb{S}$ を満たすプロキシ鍵 $\mathsf{pxk}_{\Gamma_{ID001}, KID:=1}$ を取り出す．そして，**Trans**

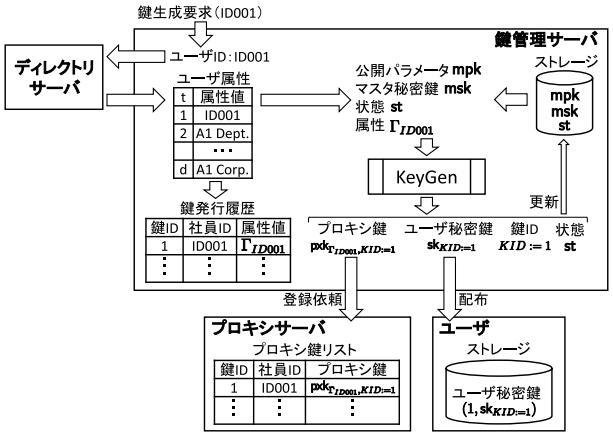


図 7 プロキシ鍵とユーザ秘密鍵生成の流れ
Fig. 7 The flow for proxy key and user private key generation.

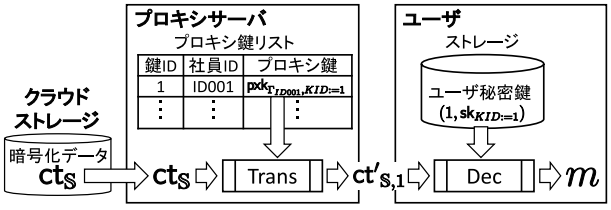


図 8 暗号化データの取得・復号の流れ
Fig. 8 The flow for decrypting the encrypted data.

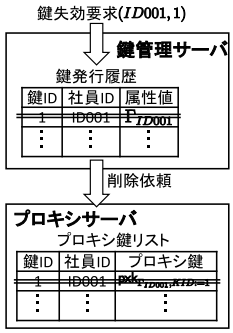


図 9 ユーザ秘密鍵の失効の流れ
Fig. 9 The flow for revoking the user private key.

アルゴリズムを実行し，部分復号データ $\mathsf{ct}'_{KID:=1}$ を得る．この部分復号データを，利用したプロキシ鍵に関連付けた鍵 ID とともにユーザに送付する．最後にユーザは，鍵 ID に対応するユーザ秘密鍵 $\mathsf{sk}_{KID:=1}$ を用いて **Dec** アルゴリズムを実行し，復号結果 m を得る．

5.4 ユーザ秘密鍵の失効手順

ユーザ秘密鍵を失効させる手順を図 **9** に示す．4.1 節で述べたように，ユーザ秘密鍵を失効させるには対応するプロキシ鍵を消去するだけで十分である．そこで，鍵管理サーバはシステム管理者から失効させたいユーザ秘密鍵として，社員 ID と鍵 ID ペア $(ID001, 1)$ を受け取り，自身の管理する鍵発行履歴を参照して鍵 ID=1 のユーザ秘密鍵が有効な状態にあることを確認したのちに，履歴を無効状態に変更するとともに，プロキシサーバに対してプロキシ鍵 $\mathsf{pxk}_{\Gamma_{ID001},KID:=1}$ の削除を依頼する．削除依頼を受けたプロキシサーバは，自身の管理するプロキシ鍵リストから鍵 ID=1 のプロキシ鍵 $\mathsf{pxk}_{\Gamma_{ID001},KID:=1}$ を消去する．5.3 節で述べたように，ユーザがクラウドストレージに保管された暗号化データを復号するには，プロキシサーバが管理するプロキシ鍵 $\mathsf{pxk}_{\Gamma_{ID001},KID:=1}$ が必要であることから，これを削除することでユーザはクラウドストレージに保管された暗号化データを復号できなくなる．すなわち，ユーザ秘密鍵の失効が完了する．

6. 考察

6.1 要件への適合性

3 章で述べた要件を満たすことを考察する．

［要件 1：暗号化時に失効されたユーザがだれかを意識しないでよいこと］

失効処理は鍵管理サーバとプロキシサーバが連携して実施するため，ユーザは失効処理について意識する必要がないし，だれが失効しているかを意識する必要もない．そのため，暗号化においては復号できるユーザの条件を指定するだけでよく，だれが失効しているかを意識する必要がない．

［要件 2：復号できていた暗号化データが復号できなくなること］

暗号化データの復号には，プロキシ鍵とユーザ秘密鍵のペアが必要であることから，プロキシ鍵を削除することで，その瞬間から対応するユーザ秘密鍵ではクラウドストレージに保管された暗号化データの復号が行えなくなる．すなわち，ユーザ秘密鍵を失効して，復号権限を無効化できる．プロキシサーバは，企業や委託先で管理されているサーバであることから，ユーザ秘密鍵を削除するより容易に実施できる．また，ユーザ秘密鍵が漏洩した際は，従来と同一の属性集合もしくはアクセス構造でユーザ秘密鍵を再発行することになるが，この場合も新しく生成した乱数からユーザ秘密鍵とプロキシ鍵のペアを生成することから，従来のユーザ秘密鍵は無効にしたまま，新しいユーザ秘密鍵を再発行することができる．

なお，我々は暗号化データはつねにクラウドストレージに保管され，復号者がつねにクラウドストレージから暗号化データをダウンロードして閲覧することを仮定している．この場合はプロキシ鍵の削除により，対応するユーザ秘密鍵が無効化できるが，プロキシ鍵で部分復号された部分復号結果が復号者端末に保管された場合，対応するプロキシ鍵を削除しても，部分復号結果はユーザ秘密鍵で復号可能なままとなるという制約がある．

［要件 3：失効にともない，クラウドストレージ上の暗号化データの更新処理が不要もしくは軽微であること］

本方式では，プロキシ鍵を削除するだけで失効ができる．そのため，公開鍵の更新や，クラウドストレージ上の暗号化データの再暗号化などはいっさい不要で，失効処理にともなって新しくクラウドストレージ上の暗号化データに対する処理は追加されていない．なお，システム全体で考えた場合，新たに設けたプロキシサーバでの計算量が増えたように見えるが，これは従来端末側で実施していた復号処理の一部がプロキシサーバで行われるだけで，システム全体での計算量の増加はない．詳細は 6.2 節で考察する．

［要件 4：失効していないユーザ秘密鍵は従来どおり使えること］

本方式では，失効したいユーザ秘密鍵に対応するプロキシ鍵を削除するだけで失効ができる．そのため，他ユーザのユーザ秘密鍵はいっさい更新不要であり，従来どおり利用することができる．

6.2 計算量

本方式では，関数型暗号の復号鍵を要素ごとに分解してプロキシ鍵とユーザ秘密鍵に分割し，復号処理時はプロキシ鍵での部分復号と，ユーザ秘密鍵による最終復号の 2 段階を踏むように変更しただけであるため，計算量はオリジナルの関数型暗号と同一である．ここでは，鍵ポリシ型を例に，その計算量の変化について考察する．

［**KeyGen アルゴリズムの計算量**］

4.2.2 項で述べたように，**KeyGen** アルゴリズムでは，関数型暗号の復号鍵を生成し，それを分割してプロキシ鍵とユーザ秘密鍵にする．そのため，計算量はオリジナルの関数型暗号と同一である．失効機能を付け加えることによる新たな演算はなく，計算量の増加はない．

［**Trans アルゴリズムの計算量**］

オリジナルの関数型暗号 **Dec** アルゴリズムの処理は，本提案方式の **Trans** アルゴリズムと **Dec** アルゴリズムに分割されているが，トータルの計算量は同じである．**Trans** アルゴリズムは，関数型暗号 **Dec** アルゴリズムとほぼ同一であるが，ベクトル空間 $\mathbb{V}$ におけるペアリング演算 1 回（K_1 の計算），群 G_T 上での K_1 と K_2 の積，および c_{d+1}/K の除算が **Dec** アルゴリズムに移行しているので，その分だけ計算量は少なくなる．失効機能を付け加えることによる新たな演算はなく，計算量の増加はない．

［**Dec アルゴリズムの計算量**］

上記 **Trans** アルゴリズムで述べたが，**Dec** アルゴリズムではベクトル空間 $\mathbb{V}$ におけるペアリング演算 1 回（K_1 の計算），群 G_T 上での K_1 と K_2 の積，および c_{d+1}/K の除算だけを行う必要がある．失効機能を付け加えることによる新たな演算はなく，計算量の増加はない．

逆に，Green らの方式 [8] と同様に，属性に依存した処理はすべて **Trans** アルゴリズムで行われるため，オリジナルの Takashima らの方式 [18] に比べ，復号者端末での処理は大きく低減できることが期待される．そのため，パソコンやスマートフォン，よりリソースの限られる IoT デバイスへの適用も期待できる．

7. まとめ

本論文では，Takashima ら [18] が提案した関数型暗号において，復号鍵を失効させるための仕組みについて提案した．具体的には，プロキシ支援型のアプローチを採用し，プロキシサーバとユーザが協調して復号処理を行うようにすることで，プロキシサーバが管理するプロキシ鍵を削除するだけでユーザ秘密鍵を失効できる仕組みを実現した．これにより，暗号化する際は失効しているユーザがだれかを意識しなくてよいため要件 1 を満たし，さらに失効処理を行った時点で過去に復号できていた暗号化データをすぐに復号できないようにする失効が実現できるため要件 2 を満たす．また，クラウドストレージにおいて暗号化データの再暗号化などの処理がいっさい不要であることから要件 3 を満たし，失効していないユーザにはまったく影響が出ないことから要件 4 も満たす．また，関数型暗号の復号鍵を単純に 2 つに分けるというシンプルな手法で，プロキシ鍵とユーザ秘密鍵を生成するようにしたことで計算量の増加はない．さらに，鍵ポリシ型と暗号文ポリシ型の双方の関数型暗号に対して失効機能を付与し，かつ adaptive 状況下で安全性証明が付けられることも示した．

今後は，本手法を検索可能暗号に応用し，検索可能暗号の検索鍵を失効する仕組みについて実現を目指していく．

参考文献

[1] Attrapadung, N., Furukawa, J. and Imai, H.: Forward-secure and searchable broadcast encryption with short ciphertexts and private keys, *ASIACRYPT 2006*, LNCS, Vol.4284, pp.161–177, Springer, Heidelberg (2006).

[2] Attrapadung, N. and Imai, H.: Attribute-Based Encryption Supporting Direct/Indirect Revocation Modes, *Cryptography and Coding*, Parker, M.G. (Ed.), pp.278–300, Springer Berlin Heidelberg (2009).

[3] Bethencourt, J., Sahai, A. and Waters, B.: Ciphertext-Policy Attribute-Based Encryption, *2007 IEEE Symposium on Security and Privacy* (*SP '07*), pp.321–334 (2007).

[4] Datta, P., Dutta, R. and Mukhopadhyay, S.: Adaptively Secure Unrestricted Attribute-Based Encryption with Subset Difference Revocation in Bilinear Groups of Prime Order, *Progress in Cryptology – AFRICACRYPT 2016*, Pointcheval, D., Nitaj, A. and Rachidi, T. (Eds.), pp.325–345, Springer International Publishing (2016).

[5] Emura, K., Takayasu, A. and Watanabe, Y.: Efficient identity-based encryption with Hierarchical key-insulation from HIBE, Designs, Codes and Cryptography, Vol.89, pp.2397–2431, Springer Berlin Heidelberg (2021).

[6] Fan, K., Wang, J., Wang, X. and Yang, Y.: Proxy-assisted access control scheme of cloud data for smart cities, *Personal and Ubiquitous Computing*, pp.937–947, Springer Berlin Heidelberg (2017).

[7] Goyal, V., Jain, A., Pandey, O. and Sahai, A.: Bounded ciphertext policy attribute-based encryption, *ICALP 2008*, LNCS, Vol.5126, pp.579–591, Springer, Heidelberg (2008).

[8] Green, M., Hohenberger, S. and Waters, B.: Outsourcing the Decryption of ABE Ciphertexts, *USENIX 2011* (2011).

[9] Hu, Z., Liu, S., Chen, K. and Liu, J.K.: Revocable Identity-Based Encryption from the Computational Diffie-Hellman Problem, *ACISP 2018*, LNCS, Vol.10946, pp.265–283, Springer, Heidelberg (2018).

[10] Kawai, Y. and Takashima, K.: Fully-Anonymous Functional Proxy-Re-Encryption, Cryptology ePrint Archive, Report 2013/318 (2013).

[11] Lai, J., Mu, Y., Guo, F., Susilo, W. and Chen, R.: Anonymous Identity-Based Broadcast Encryption with Revocation for File Sharing, *ACISP 2016*, LNCS, Vol.9723, pp.223–239, Springer, Heidelberg (2016).

[12] Lee, K.: Revocable Hierarchical Identity-Based Encryption with Adaptive Security, Cryptology ePrint Archive, Report 2016/749 (2016).

[13] 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.), pp.235–254, Springer Berlin Heidelberg (2013).
[14] Lewko, A., Okamoto, T., Sahai, A., Takashima, K. and Waters, B.: Fully secure functional encryption: Attribte-based encryption and (hierarchical) inner product encryption, *EUROCRYPT 2010*, LNCS, Vol.6110, pp.62–91, Springer, Heidelberg (2010).
[15] Li, J., Yao, W., Zhang, Y., Qian, H. and Han, J.: Flexible and Fine-Grained Attribute-Based Data Storage in Cloud Computing, *IEEE Trans. Services Computing*, Vol.10, No.5, pp.785–796 (2017).
[16] Li, X., Tang, S., Xu, L., Wang, H. and Chen, J.: Two-Factor Data Access Control With Efficient Revocation for Multi-Authority Cloud Storage Systems, *IEEE Access*, Vol.5, pp.393–405 (2017).
[17] Nomura, K., Mohri, M., Shiraishi, Y. and Morii, M.: Attribute Revocable Multi-Authority Attribute-Based Encryption with Forward Secrecy for Cloud Storage, *IEICE Trans. Information and Systems*, Vol.E100-D, No.10, pp.2420–2431 (2017).
[18] Okamoto, T. and Takashima, K.: Fully secure functional encryption with general relations from the decisional linear assumption, *CRYPTO 2010*, Rabin, T. (Ed.), LNCS, Vol.6223, pp.191–208, Springer, Heidelberg (2010).
[19] Okamoto, T. and Takashima, K.: Fully Secure Unbounded Inner-Product and Attribute-Based Encryption, *Advances in Cryptology – ASIACRYPT 2012*, Wang, X. and Sako, K. (Eds.), pp.349–366, Springer Berlin Heidelberg (2012).
[20] Sahai, A., Seyalioglu, H. and Waters, B.: Dynamic Credentials and Ciphertext Delegation for Attribute-Based Encryption, *Advances in Cryptology – CRYPTO 2012*, Safavi-Naini, R. and Canetti, R. (Eds.), pp.199–217, Springer Berlin Heidelberg (2012).
[21] Seo, J.H. and Emura, K.: Adaptive-ID Secure Revocable Hierarchical Identity-Based Encryption, *IWSEC 2015*, LNCS, Vol.9241, pp.21–38, Springer, Heidelberg (2015).
[22] Sun, Y., Zhang, F. and Fu, A.: Revocable Certificateless Encryption with Ciphertext Evolution, *ACISP 2018*, LNCS, Vol.10946, pp.741–749, Springer, Heidelberg (2018).
[23] Takayasu, A. and Watanabe, Y.: Lattice-Based Revocable Identity-Based Encryption with Bounded Decryption Key Exposure Resistance, *ACISP 2017*, LNCS, Vol.10342, pp.184–204, Springer, Heidelberg (2017).
[24] Wang, G., Liu, Q. and Wu, J.: Hierarchical attribute-based encryption for fine-grained access control in cloud storage services, *ACM CCS'10*, pp.735–737 (2010).
[25] Waters, B.: Ciphertext-policy attribute-based encryption: An expressive, efficient, and provably secure realization, *PKC 2011*, LNCS, Vol.6571, pp.53–70, Springer, Heidelberg (2011).
[26] Wu, Q., Qin, B., Zhang, L., Domingo-Ferrer, J. and Farràs, O.: Bridging Broadcast Encryption and Group Key Agreement, *ASIACRYPT 2011*, pp.143–160, Springer Berlin Heidelberg (2011).
[27] Xu, S., Yang, G., Mu, Y. and Susilo, W.: Mergeable and Revocable Identity-Based Encryption, *ACISP 2017*, LNCS, Vol.10342, pp.147–167, Springer, Heidelberg (2017).
[28] Yamada, K., Attrapadung, N., Emura, K., Hanaoka, G. and Tanaka, K.: Generic Constructions for Fully Secure Revocable Attribute-Based Encryption, *Computer Security – ESORICS 2017*, Foley, S.N., Gollmann, D. and Snekkenes, E. (Eds.), pp.532–551, Springer International Publishing (2017).
[29] Yang, K. and Jia, X.: Expressive, Efficient, and Revocable Data Access Control for Multi-Authority Cloud Storage, *IEEE Trans. Parallel and Distributed Systems*, Vol.25, No.7, pp.1735–1744 (2014).
[30] Yang, Y., Liu, J., Wei, Z. and Huang, X.: Towards Revocable Fine-Grained Encryption of Cloud Data: Reducing Trust upon Cloud, *Information Security and Privacy*, pp.127–144, Springer International Publishing (2017).
[31] Yang, Y., Liu, J.K., Liang, K., Choo, K.-K.R. and Zhou, J.: Extended Proxy-Assisted Approach: Achieving Revocable Fine-Grained Encryption of Cloud Data, *ESORICS 2015*, pp.146–166, Springer International Publishing (2015).
[32] Yu, P., Wen, Q., Ni, W., Li, W., Sun, C., Zhang, H. and Jin, Z.: Decentralized, Revocable and Verifiable Attribute-Based Encryption in Hybrid Cloud System, *Wireless Personal Communications*, Vol.106, pp.719–738, Springer Berlin Heidelberg (2019).
[33] Yu, S., Wang, C., Ren, K. and Lou, W.: Achieving Secure, Scalable, and Fine-grained Data Access Control in Cloud Computing, *IEEE INFOCOM 2010*, pp.1–9 (2010).
[34] 森 拓海, 川合 豊, 松田 規：安全に人事異動・組織変更に対応する関数型暗号システムの検討，*DICOMO 2015*, Vol.2015, pp.1582–1588 (2015).

付 録

A.1 Theorem 1 証明で利用する Lemma

Theorem 1 で引用した Lemma を下記に示す．

Lemma 3. 任意の確率的多項式時間攻撃者 $\mathcal{A}$ に対して，$|\mathsf{Adv}^{(0)}_{\mathcal{A}}(\lambda) - \mathsf{Adv}^{(1)}_{\mathcal{A}}(\lambda)| \leq \mathsf{Adv}^{\mathsf{P1}}_{\mathcal{B}_1}(\lambda)$ となる計算量仮定 *Problem 1* に対する確率的多項式時間攻撃者 $\mathcal{B}_1$ が存在する．

Proof. 攻撃者 $\mathcal{A}$ を利用して，Problem 1 を解く確率的アルゴリズム $\mathcal{B}_1$ を下記のように構成する．

(1) $\mathcal{B}_1$ は Problem 1 インスタンス $(\mathbf{param}_{\overrightarrow{n}}, \mathbb{B}_0, \hat{\mathbb{B}^*_0}, \boldsymbol{e}_{\beta,0}, \{\mathbb{B}_t, \hat{\mathbb{B}^*_t}, \boldsymbol{e}_{\beta,t,1}, \boldsymbol{e}_{t,j}\}_{t=1,\ldots,d;j=2,\ldots,n_t})$ を受け取る．

(2) $\mathcal{B}_1$ は，Problem 1 インスタンスから公開パラメータ $\mathsf{mpk} := (1^\lambda, \mathsf{param}_{\overrightarrow{n}}, \{\hat{\mathbb{B}}_t\}_{t=0,\ldots,d})$ を計算し，攻撃者 $\mathcal{A}$ に与える．ここで，$\hat{\mathbb{B}}_0 := (\boldsymbol{b}_{0,1}, \boldsymbol{b}_{0,3}, \boldsymbol{b}_{0,5})$, $\hat{\mathbb{B}}_t := (\boldsymbol{b}_{t,1}, \ldots, \boldsymbol{b}_{t,n_t}, \boldsymbol{b}_{t,3n_t+1})$ である．

(3) 攻撃者 $\mathcal{A}$ からアクセス構造 $\mathbb{S}$ に対する鍵生成クエリを受け取ったら，Problem 1 インスタンスに含まれる基底 $\{\hat{\mathbb{B}^*_t}\}_{t=0,\ldots,d}$ を用いて Game0 に記載の手順でプロキシ鍵 $\mathsf{pxk}_{\mathbb{S}_\iota, KID_\iota}$ とユーザ秘密鍵 sk_{KID_ι} を生成する．CorruptProxyKey クエリではプロキシ鍵 $\mathsf{pxk}_{\mathbb{S}_\iota, KID_\iota}$ を，CorruptSecretKey クエリではユーザ秘密鍵 sk_{KID_ι} を攻撃者 $\mathcal{A}$ に与える．

(4) 攻撃者 $\mathcal{A}$ からチャレンジメッセージ $(m^{(0)}, m^{(1)})$，属性集合 $\Gamma^* := \{(t, \overrightarrow{x}_t) | 1 \leq t \leq d\}$ を受け取ったら，下

記のようにチャレンジ暗号文を生成し，攻撃者 $\mathcal{A}$ に送付する．

$\zeta \xleftarrow{\cup} \mathbb{F}_q, b \xleftarrow{\cup} \{0,1\}, \boldsymbol{c}_0 := \boldsymbol{e}_{\beta,0} + \zeta \boldsymbol{b}_{0,3}$,

$\boldsymbol{c}_t := x_{t,1}\boldsymbol{e}_{\beta,t,1} + \sum_{j=2}^{n_t} x_{t,j}\boldsymbol{e}_{t,j}, c_{d+1} := g_T^{\zeta} m^{(b)}$,

(5) 攻撃者 $\mathcal{A}$ からクエリを受け取ったら，ステップ (3) と同様に応答する．

(6) 攻撃者 $\mathcal{A}$ がビット b' を出力したら，$\mathcal{B}_1$ は $b = b'$ の場合は 1 を，そうでなければ 0 を出力する．

Takashima ら [18] により，確率的アルゴリズム $\mathcal{B}_1$ は，Problem 1 インスタンスにおいて $\beta = 0$ であれば Game0 の復号鍵とチャレンジ暗号文の分布が同一であること，また $\beta = 1$ であれば Game1 と分布が同一であることが示されているため，本ゲームのプロキシ鍵，ユーザ秘密鍵，チャレンジ暗号文の分布も Game0（$\beta = 0$）と Game1（$\beta = 1$）と分布は同一となる．よって，2 つのゲームを識別することは Problem 1 インスタンスを識別することと同等である． □

Lemma 4. 任意の確率的多項式時間攻撃者 $\mathcal{A}$ に対して，$|\mathsf{Adv}_{\mathcal{A}}^{(2-h)}(\lambda) - \mathsf{Adv}_{\mathcal{A}}^{(2-h^+)}(\lambda)| \leq \mathsf{Adv}_{\mathcal{B}_{2,h}^+}^{\mathsf{P2}}(\lambda) + (d+3)/q$ となる計算量仮定 *Problem 2* に対する確率的多項式時間攻撃者 $\mathcal{B}_2^+$ が存在する．

Proof. 攻撃者 $\mathcal{A}$ を利用して，Problem 2 を解く確率的アルゴリズム $\mathcal{B}_2^+$ を下記のように構成する．

(1) $\mathcal{B}_2^+$ は Problem 2 インスタンス $(\mathbf{param}_{\overrightarrow{n}}, \hat{\mathbb{B}}_0, \mathbb{B}_0^*, \boldsymbol{h}_{\beta,0}^*, \boldsymbol{e}_0, \{\hat{\mathbb{B}}_t, \mathbb{B}_t^*, \boldsymbol{h}_{\beta,t,j}, \boldsymbol{e}_{t,j}\}_{t=1,\ldots,d;j=2,\ldots,n_t})$ を受け取る．

(2) $\mathcal{B}_2^+$ は，Problem 2 インスタンスから公開パラメータ $\mathsf{mpk} := (1^\lambda, \mathsf{param}_{\overrightarrow{n}}, \{\hat{\mathbb{B}'}_t\}_{t=0,\ldots,d})$ を計算し，攻撃者 $\mathcal{A}$ に与える．ここで，$\hat{\mathbb{B}'}_0 := (\boldsymbol{b}_{0,1}, \boldsymbol{b}_{0,3}, \boldsymbol{b}_{0,5})$, $\hat{\mathbb{B}'}_t := (\boldsymbol{b}_{t,1}, \ldots, \boldsymbol{b}_{t,n_t}, \boldsymbol{b}_{t,3n_t+1})$ である．

(3) 攻撃者 $\mathcal{A}$ からアクセス構造 $\mathbb{S}$ に対する ι 番目の鍵生成クエリを受け取ったら，下記のようにプロキシ鍵 $\mathsf{pxk}_{\mathbb{S}_\iota, KID_\iota}$ とユーザ秘密鍵 sk_{KID_ι} を生成する．

(a) $1 \leq \iota \leq h$ の場合，Problem 2 インスタンスに含まれる $\{\mathbb{B}_t^*\}_{t=0,\ldots,d}$ を用いて，図 5 の Game2-$(h+1)$ に示した手順で生成する．

(b) $\iota = h+1$ の場合，Problem 2 インスタンスを用いて下記のように生成する．ここで，$M = (M_{i,k})_{i=1,\ldots,\ell;k=1,\ldots,r}$ とする．

$\pi_t, \mu_t, g_k, \tilde{\mu}_k \xleftarrow{\cup} \mathbb{F}_q$ for $t = 1,\ldots,d; k = 1,\ldots,r$,

$\tilde{\boldsymbol{p}}_{\beta,0}^* := \sum_{k=1}^{r} \left(g_k \boldsymbol{h}_{\beta,0}^* + \tilde{\mu}_k \boldsymbol{b}_{0,1}^*\right)$,

for $t = 1,\ldots,d; k = 1,\ldots,r; j = 1,\ldots,n_t$;

$\boldsymbol{p}_{\beta,t,j}^* := \pi_t \boldsymbol{h}_{\beta,t,j}^* + \mu_t \boldsymbol{b}_{t,j}^*$,

$\tilde{\boldsymbol{p}}_{\beta,t,k,j}^* := g_k \boldsymbol{h}_{\beta,t,j}^* + \tilde{\mu}_k \boldsymbol{b}_{t,j}^*$,

$\boldsymbol{k}_0^* := -\tilde{\boldsymbol{p}}_{\beta,0}^* + \boldsymbol{b}_{0,3}^*$,

for $i = 1,\ldots,\ell$;

if $\rho(i) = (t, \overrightarrow{v}_i)$,

$\boldsymbol{k}_i^* := \sum_{j=1}^{n_t} v_{i,j} \boldsymbol{p}_{\beta,t,j}^* + \sum_{k=1}^{r} M_{i,k} \tilde{\boldsymbol{p}}_{\beta,t,k,1}^*$,

if $\rho(i) = \neg(t, \overrightarrow{v}_i)$,

$\boldsymbol{k}_i^* := \sum_{j=1}^{n_t} v_{i,j} \left(\sum_{k=1}^{r} M_{i,k} \tilde{\boldsymbol{p}}_{\beta,t,k,j}^*\right)$,

(c) $\iota \geq h+2$ の場合，Problem 2 インスタンスの $\{\mathbb{B}_t^*\}_{t=0,\ldots,d}$ を用いて，図 5 の Game1 に示した手順で生成する．

CorruptProxyKey クエリではプロキシ鍵 $\mathsf{pxk}_{\mathbb{S}_\iota, KID_\iota}$ を，CorruptSecretKey クエリではユーザ秘密鍵 sk_{KID_ι} を攻撃者 $\mathcal{A}$ に与える．

(4) 攻撃者 $\mathcal{A}$ からチャレンジメッセージ $(m^{(0)}, m^{(1)})$，属性集合 $\Gamma^* := \{(t, \overrightarrow{x}_t) | 1 \leq t \leq d\}$ を受け取ったら，下記のようにチャレンジ暗号文を生成し，攻撃者 $\mathcal{A}$ に送付する．

$\zeta \xleftarrow{\cup} \mathbb{F}_q, b \xleftarrow{\cup} \{0,1\}, \boldsymbol{q}_0 \xleftarrow{\cup} \mathsf{span}\langle \boldsymbol{b}_{0,5} \rangle$,

$\boldsymbol{q}_t \xleftarrow{\cup} \mathsf{span}\langle \boldsymbol{b}_{t,3n_t+1} \rangle, \boldsymbol{c}_0 := \boldsymbol{e}_0 + \zeta \boldsymbol{b}_{0,3} + \boldsymbol{q}_0$,

$\boldsymbol{c}_t := \sum_{j=1}^{n_t} x_{t,j} \boldsymbol{e}_{t,j} + \boldsymbol{q}_t, c_{d+1} := g_T^{\zeta} m^{(b)}$.

(5) 攻撃者 $\mathcal{A}$ からクエリを受け取ったら，ステップ (3) と同様に応答する．

(6) 攻撃者 $\mathcal{A}$ がビット b' を出力したら，$\mathcal{B}_2^+$ は $b = b'$ の場合は 1 を，そうでなければ 0 を出力する．

なお，ステップ (3)-(b) の $\tilde{\boldsymbol{p}}_{\beta,0}^*$，$\boldsymbol{p}_{\beta,t,j}^*$，$\tilde{\boldsymbol{p}}_{\beta,t,k,j}^*$ は，$\theta_t := \pi_t \delta + \mu_t$，$f_k := g_k \delta + \tilde{\mu}_k$，$s_0 := \sum_{k=1}^{r} f_k$，$a_0 := \sum_{k=1}^{r} g_k$ としたとき，下記のように計算できる．

$\tilde{\boldsymbol{p}}_{0,0}^* = (s_0, 0, 0, a_0\delta_0, 0)_{\mathbb{B}_0^*}, \tilde{\boldsymbol{p}}_{1,0}^* = (s_0, a_0 u_0, 0, a_0\delta_0, 0)_{\mathbb{B}_0^*}$,

$\boldsymbol{p}_{0,t,j}^* := (\theta_t \overrightarrow{e}_{t,j}, 0^{n_t}, \pi_t \overrightarrow{\delta}_{t,j}, 0)_{\mathbb{B}_t^*}$,

$\tilde{\boldsymbol{p}}_{0,t,k,j}^* := (f_k \overrightarrow{e}_{t,j}, 0^{n_t}, g_k \overrightarrow{\delta}_{t,j}, 0)_{\mathbb{B}_t^*}$,

$\boldsymbol{p}_{1,t,j}^* := (\theta_t \overrightarrow{e}_{t,j}, \pi_t \overrightarrow{u}_{t,j}, \pi_t \overrightarrow{\delta}_{t,j}, 0)_{\mathbb{B}_t^*}$,

$\tilde{\boldsymbol{p}}_{1,t,k,j}^* := (f_k \overrightarrow{e}_{t,j}, g_k \overrightarrow{u}_{t,j}, g_k \overrightarrow{\delta}_{t,j}, 0)_{\mathbb{B}_t^*}$

ここで，δ，δ_0，$\overrightarrow{e}_{t,j}$，$\overrightarrow{u}_{t,j}$，$\overrightarrow{\delta}_{t,j}$ は Problem 2 で定義された値であり，δ，δ_0，$\overrightarrow{\delta}_{t,j}$ は独立して一様分布する．さらに，π_t，μ_t，g_k，$\tilde{\mu}_k$ は $\mathbb{F}_q$ から独立して一様分布に選んでおり，$\{\theta_t, \pi_t\}_{t=1,\ldots,d}$, $\{f_k, g_k\}_{k=1,\ldots,r}$ もそれぞれ独立に一様分布となる．

$\beta = 0$ の場合，プロキシ鍵とユーザ秘密鍵は下記のように計算される．

$\boldsymbol{k}_0^* = (-s_0, 0, 1, -a_0\delta_0, 0)_{\mathbb{B}_0^*}$,

for $i = 1,\ldots,\ell$;

if $\rho(i) = (t, \overrightarrow{v}_i)$,

$\boldsymbol{k}_i^* = (s_i \overrightarrow{e}_{t,1} + \theta_t \overrightarrow{v}_i, 0^{n_t}, \overrightarrow{\eta}_i, 0)_{\mathbb{B}_t^*}$,

if $\rho(i) = \neg(t, \overrightarrow{v}_i)$,

$\boldsymbol{k}_i^* = (s_i \overrightarrow{v}_i, 0^{n_t}, \overrightarrow{\eta}_i, 0)_{\mathbb{B}_t^*}$

ここで，$\overrightarrow{\eta}_i := \pi_t \sum_{j=1}^{n_t} v_{i,j} \overrightarrow{\delta}_{t,j} + (\sum_{k=1}^{r} g_k M_{i,k}) \overrightarrow{\delta}_{t,1}$, $\overrightarrow{\eta'}_i = (\sum_{k=1}^{r} M_{i,k} g_k)(\sum_{j=1}^{n_t} v_{i,j} \overrightarrow{\delta}_{t,j})$ であるが，$\overrightarrow{\delta}_{t,j}$ は $\mathbb{F}_q^{n_t}$ 上を独立して一様分布するため，$\overrightarrow{\eta}_i$，$\overrightarrow{\eta'}_i$ もそれぞれ

$\mathbb{F}_q^{n_t}$ 上を独立して一様分布する．また，δ_0，θ_t も $\mathbb{F}_q$ 上を独立して一様分布することから，各変数の分布も含めて正しく Game 2-h の秘密鍵をシミュレートしている．そのため，Takashima らが示したのと同様に確率 $(d+2)/q$ の例外を除いて，正しく Game2-h のチャレンジ暗号文，プロキシ鍵，ユーザ秘密鍵がシミュレートできる．

$\beta = 1$ の場合，プロキシ鍵とユーザ秘密鍵は下記のように計算される．

$$\boldsymbol{k}_0^* = (-s_0, -w_0, 1, -a_0\delta_0, 0)_{\mathbb{B}_0^*},$$
for $i = 1, \ldots, \ell$;
if $\rho(i) = (t, \overrightarrow{v}_i)$,
$$\boldsymbol{k}_i^* = (s_i \overrightarrow{e}_{t,1} + \theta_t \overrightarrow{v}_i, (\pi_t \overrightarrow{v}_i + a_i \overrightarrow{e}_{t,1}) \cdot Z_t, \overrightarrow{\eta}_i, 0)_{\mathbb{B}_t^*},$$
if $\rho(i) = \neg(t, \overrightarrow{v}_i)$,
$$\boldsymbol{k}_i^* = (s_i \overrightarrow{v}_i, a_i \overrightarrow{v}_i \cdot Z_t, \overrightarrow{\eta'}_i, 0)_{\mathbb{B}_t^*}$$

ここで，$\beta = 0$ のケースとの違いは，各 $\boldsymbol{k}_i^*$ の第 2 成分であり，それ以外の要素については $\beta = 0$ のケースと同じであることから正しくシミュレートできている．Takashima らは，暗号文の要素 $\boldsymbol{c}_0$ の第 2 要素 r_0 と，秘密鍵の要素 $\boldsymbol{k}_0^*$ の第 2 要素 w_0 に $w_0 := a_0/r_0$ という関係性があるため，a_0 の独立性を確認するため，$a_0 = \overrightarrow{1} \cdot \overrightarrow{g}^T$，$(a_1, \ldots, a_\ell)^T = M \cdot \overrightarrow{g}^T$，$U_t = (Z_t^{-1})^T$ という関係があることから，$\overrightarrow{w}_i := (\pi_t \overrightarrow{v}_i + a_i \overrightarrow{e}_{t,1}) \cdot Z_t$，$\overrightarrow{\tilde{w}}_i := a_i \overrightarrow{v}_i \cdot Z_t$，$\overrightarrow{r}_t := \overrightarrow{x}_t \cdot U_t$ の独立性の検証が必要となると指摘していた．そこで下記の 5 パターンについて確認する．

(1) $\gamma(i) = 1$ で $[\rho(i) = (t, \overrightarrow{v}_i) \wedge (t, \overrightarrow{x}_t) \in \Gamma \wedge \overrightarrow{v}_i \cdot \overrightarrow{x}_t = 0]$
(2) $\gamma(i) = 1$ で $[\rho(i) = \neg(t, \overrightarrow{v}_i) \wedge (t, \overrightarrow{x}_t) \in \Gamma \wedge \overrightarrow{v}_i \cdot \overrightarrow{x}_t \neq 0]$
(3) $\gamma(i) = 0$ で $[\rho(i) = (t, \overrightarrow{v}_i) \wedge (t, \overrightarrow{x}_t) \in \Gamma]$
(4) $\gamma(i) = 0$ で $[\rho(i) = \neg(t, \overrightarrow{v}_i) \wedge (t, \overrightarrow{x}_t) \in \Gamma]$
(5) $[\rho(i) = (t, \overrightarrow{v}_i) \wedge (t, \overrightarrow{x}_t) \notin \Gamma]$ もしくは $[\rho(i) = \neg(t, \overrightarrow{v}_i) \wedge (t, \overrightarrow{x}_t) \notin \Gamma]$

Takashima らが示したように，チャレンジ暗号文を復号できない場合は上記のケース 3〜5 の場合に該当し，$\overrightarrow{w}_i$，$\overrightarrow{\tilde{w}}_i$，$\overrightarrow{r}_t$ は a_0 とは独立に一様分布となる．そのため，Problem 2 の δ が 0 になる例外を除いて，チャレンジ暗号文，プロキシ鍵，ユーザ秘密鍵は正しくシミュレートできる．

一方，チャレンジ暗号文が復号できるアクセス構造が指定された場合，上記のケース (1)，(2) について考慮してプロキシ鍵とユーザ秘密鍵の分布を評価する必要がある．自明な攻撃を防ぐために，攻撃者はプロキシ鍵かユーザ秘密鍵のいずれか一方のみが取得できるように制約しているので，それぞれのケースについて考察する．

［プロキシ鍵のみを取得する場合］

攻撃者がプロキシ鍵のみ取得でき，ユーザ秘密鍵が取得できないケースについて考察する．プロキシ鍵には，$\overrightarrow{w}_i$，$\overrightarrow{\tilde{w}}_i$ が含まれるため，その分布について前述のケース (1) と (2) についての評価が必要となる．ケース (1) については，$\overrightarrow{w}_i \cdot \overrightarrow{r}_t = (\pi_t \overrightarrow{v}_i + a_i \overrightarrow{e}_{t,1}) \cdot Z_t \cdot \overrightarrow{x}_t \cdot U_t = a_i$ であることから，$(\overrightarrow{w}_i, \overrightarrow{r}_t)$ は C_{a_i} 上を独立一様分布する．同様にケース (2) についても，$\overrightarrow{w}_i \cdot \overrightarrow{r}_t = a_i(\overrightarrow{v}_i \cdot \overrightarrow{x}_t)$ であることから，$(\overrightarrow{w}_i, \overrightarrow{r}_t)$ は $C_{a_i(\overrightarrow{v}_i \cdot \overrightarrow{x}_t)}$ 上を一様分布する．これより，攻撃者はプロキシ鍵の $\overrightarrow{w}_i$，$\overrightarrow{\tilde{w}}_i$ だけから $(a_1, \ldots, a_\ell)$ に関する情報を得ることはできないが，チャレンジ暗号文の $\overrightarrow{r}_t$ と内積値をとることで $(a_1, \ldots, a_\ell)$ に関する情報が得られる．しかし，$(a_1, \ldots, a_\ell)$ に関する情報が得られても，攻撃者はチャレンジ暗号文とプロキシ鍵の correlation を生み出すユーザ秘密鍵の要素 $w_0 := a_0/r_0$ は入手できないため，これを入手できない攻撃者にとっては，チャレンジ暗号文とプロキシ鍵は独立一様分布に見える．

［ユーザ秘密鍵のみを取得する場合］

攻撃者がユーザ秘密鍵のみ取得でき，プロキシ鍵が取得できないケースについて考察する．この場合，$\boldsymbol{k}_0^*$ に含まれる $w_0 := a_0/r_0$ を得ることができるが，$a_0 = \sum g_k$ は独立一様分布な乱数であること，およびプロキシ鍵に含まれる $(a_1, \ldots, a_\ell)$ が得られないことから a_0 に関する追加の情報が得られず，攻撃者にとってはユーザ秘密鍵もチャレンジ暗号文とは独立一様分布に見えることが分かる．

以上の考察より，たとえ暗号化データを復号できる条件の下であっても，プロキシ鍵もしくはユーザ秘密鍵のいずれか一方しか入手できない攻撃者にとっては，Problem 2 の δ が 0 になる例外を除いて，正しくプロキシ鍵やユーザ秘密鍵をシミュレートできていることが分かる．よって，2 つのゲームを識別することは Problem 2 インスタンスを識別することと同等である． □

なお，上記証明においては，Takashima らが示した下記の Lemma を利用した．

Lemma 5. V を $\mathbb{F}_q^n$ 上の n 次元ベクトル空間，V^* をその双対空間とする．$p \in \mathbb{F}_q$ に対して，$C_p := \{(\overrightarrow{x}, \overrightarrow{v}) | \overrightarrow{x} \cdot \overrightarrow{v} = p\} \subset V \times V^*$ とする．また，$Z \xleftarrow{\cup} GL(n, \mathbb{F}_q), U := (Z^{-1})^T$ とする．このとき，あらゆる $(\overrightarrow{x}, \overrightarrow{v}), (\overrightarrow{r}, \overrightarrow{w}) \in C_p$ に対して，$Pr[\overrightarrow{x}U = \overrightarrow{r} \wedge \overrightarrow{v}Z = \overrightarrow{w}] = Pr[\overrightarrow{x}Z = \overrightarrow{r} \wedge \overrightarrow{v}U = \overrightarrow{w}] = 1/\sharp C_p$ となる．

Lemma 6. 任意の確率的多項式時間攻撃者 $\mathcal{A}$ に対して，$|\mathsf{Adv}_{\mathcal{A}}^{(2-h^+)}(\lambda) - \mathsf{Adv}_{\mathcal{A}}^{(2-(h+1))}(\lambda)| \leq \mathsf{Adv}_{\mathcal{B}_{2,h+1}}^{\mathsf{P2}}(\lambda) + (d+3)/q$ となる計算量仮定 *Problem 2* に対する確率的多項式時間攻撃者 $\mathcal{B}_2$ が存在する．

Proof. 攻撃者 $\mathcal{A}$ を利用して，Problem 2 を解く確率的アルゴリズム $\mathcal{B}_2$ を構成する．アルゴリズム $\mathcal{B}_2$ の動作はアルゴリズム $\mathcal{B}_2^+$ と以下の点を除いて同一である．

(1) ステップ (3) のケース (b) において，$\boldsymbol{k}_0^*$ の計算方法が下記のようになる．
$r_0' \xleftarrow{\cup} \mathbb{F}_q, \boldsymbol{k}_0^* := -\tilde{\boldsymbol{p}}_{\beta,0}^* + r_0' \boldsymbol{b}_{0,2}^* + \boldsymbol{b}_{0,3}^*$
(2) 最後のステップで，攻撃者 $\mathcal{A}$ がビット b' を出力したら，$\mathcal{B}_2$ は $b = b'$ の場合は 0 を，そうでなければ 1 を出力する．

分布に関しては，Lemma 4 と同様に確認することができる．

$\beta = 0$ の場合は，$\delta = 0$ の場合，すなわち確率 $1/q$ を除いて，Game 2-$(h+1)$ と分布が同一であることが分かる．

$\beta = 1$ の場合は，$\delta = 0$ の場合，および $\overrightarrow{r}_t = \overrightarrow{0}$ となる場合を除いて，Game 2-h^+ と分布が同一であることが分かる．その例外が生じる確率は $(d+2)/q$ となる． □

Lemma 7. 任意の確率的多項式時間攻撃者 $\mathcal{A}$ に対して，$\mathsf{Adv}^{(3)}_{\mathcal{A}}(\lambda) \leq \mathsf{Adv}^{(2-\nu)}_{\mathcal{A}}(\lambda) + 1/q$ である．

Proof. 新しい基底 $\mathbb{D}_0$ および $\mathbb{D}^*_0$ を定義する．

$$\theta \xleftarrow{\cup} \mathbb{F}_q, \boldsymbol{d}_{0,2} := \boldsymbol{b}_{0,2} - \theta \boldsymbol{b}_{0,3}, \boldsymbol{d}^*_{0,3} := \boldsymbol{b}^*_{0,3} + \theta \boldsymbol{b}^*_{0,2},$$
$$\mathbb{D}_0 := (\boldsymbol{b}_{0,1}, \boldsymbol{d}_{0,2}, \boldsymbol{b}_{0,3}, \boldsymbol{b}_{0,4}, \boldsymbol{b}_{0,5}),$$
$$\mathbb{D}^*_0 := (\boldsymbol{b}^*_{0,1}, \boldsymbol{b}^*_{0,2}, \boldsymbol{d}^*_{0,3}, \boldsymbol{b}^*_{0,4}, \boldsymbol{b}^*_{0,5}),$$

このとき，j 番目に生成したプロキシ鍵 $\mathsf{pxk}^{(j)}_{\mathbb{S},KID}$ の要素 $\boldsymbol{k}^{(j)*}_0$ は下記のように表せる．

$$\begin{aligned}
\boldsymbol{k}^{(j)*}_0 &= (-s^{(j)}_0, w^{(j)}_0, 1, \eta^{(j)}_0, 0)_{\mathbb{B}^*_0} \\
&= (-s^{(j)}_0, w^{(j)}_0 + \theta, 1, \eta^{(j)}_0, 0)_{\mathbb{D}^*_0} \\
&= (-s^{(j)}_0, \vartheta^{(j)}_0, 1, \eta^{(j)}_0, 0)_{\mathbb{D}^*_0} \\
\boldsymbol{c}_0 &= (\omega, r_0, \zeta, 0, \phi_0)_{\mathbb{B}_0} \\
&= (\omega, r_0, \zeta + r_0\theta, 0, \phi_0)_{\mathbb{D}_0} = (\omega, r_0, \zeta', 0, \phi_0)_{\mathbb{D}_0}
\end{aligned}$$

ただし，$\vartheta^{(j)}_0 := w^{(j)}_0 + \theta$, $\zeta' := \zeta + r_0\theta$ である．

攻撃者のビューからすれば，基底 $\mathbb{B}_0$ も基底 $\mathbb{D}_0$ も公開パラメータ $\mathsf{mpk} := (1^\lambda, \mathsf{param}_{\overrightarrow{n}}, \{\hat{\mathbb{B}}_t\}_{t=0,\ldots,d})$ と合致するため，どちらの基底で作られた暗号文でも区別はつかない．そのため，Game2-ν と Game3 は，乱数 $r_0 = 0$ の場合を除いて conceptual change である． □

Lemma 8. 任意の確率的多項式時間攻撃者 $\mathcal{A}$ に対して，$\mathsf{Adv}^{(3)}_{\mathcal{A}}(\lambda) = 0$ である．

Proof. ζ と ζ' は独立した値であることから，攻撃者はメッセージに関する情報は得られない．ゆえに，$\mathsf{Adv}^{(3)}_{\mathcal{A}}(\lambda) = 0$ が成り立つ． □

A.2 Theorem 2 の証明

各 Lemma の証明に関しては割愛するが，Takashima ら [18] が示した証明に対して，付録 A.1 で示した修正と同様の修正を行うことで証明が可能である．ただし，暗号文ポリシ型はもう一工夫が必要である．Lemma 10 の証明で，correlation を起こす 3 要素のうち w_0 および $\{a_i\}_{i=1,\cdots,\ell}$ の 2 要素がチャレンジ暗号文に埋め込まれ，残りの 1 要素 r_0 がユーザ秘密鍵に埋め込まれるため，攻撃者がチャレンジ暗号文を復号可能な属性集合を指定してユーザ秘密鍵をクエリすると，correlation に気がつく可能性がある．しかし実際は，チャレンジ暗号文に埋め込まれた $\{a_i\}_{i=1,\cdots,\ell}$ は，$\overrightarrow{w}_i := (a_i \overrightarrow{e}_{t,1} + \pi_i \overrightarrow{v}_i) \cdot Z_t$ もしくは $\overrightarrow{w}_i := a_i \overrightarrow{v}_i \cdot Z_t$ という形で埋め込まれており，行列 Z_t が $GL(n_t, \mathbb{F}_q)$ から独立一様分布で選ばれているため，チャレンジ暗号文中の $\overrightarrow{w}_i$，$\overrightarrow{w}_i$ は a_i に依存せずに $\mathbb{F}^{n_t}_q$ 上を一様分布する．これは，$\overrightarrow{w}_i$，$\overrightarrow{w}_i$ から a_i に関する情報を得るためには，対になる $\overrightarrow{x}_t U_t$ が必要になることを意味する．そこで，対になる要素 $\overrightarrow{x}_t U_t$ がプロキシ鍵に埋め込まれているため，前述の correlation が攻撃者のビューでは見えなくなることを利用して安全性証明を行う．

はじめに，Theorem 2 の証明を下記に示す．

Proof. Theorem 2 を証明するためのゲーム列を**図 A·1** に示す．Game0 が Definition 15 に示したオリジナルのゲームであり，この要素を徐々に変えていき，Game3 まで変化させる．前のゲームから変化する点は，四角枠で囲った部分である．Game0，1，2-h，2-h^+，3 の攻撃者のアドバンテージを $\mathsf{Adv}^{(0)}_{\mathcal{A}}(\lambda)$，$\mathsf{Adv}^{(1)}_{\mathcal{A}}(\lambda)$，$\mathsf{Adv}^{(2-h)}_{\mathcal{A}}(\lambda)$，$\mathsf{Adv}^{(2-h^+)}_{\mathcal{A}}(\lambda)$，$\mathsf{Adv}^{(3)}_{\mathcal{A}}(\lambda)$ とする．このとき，それぞれのアドバンテージの違いは後続の Lemma 9，10，11，12，13 のように計算することができるため，$\mathsf{Adv}^{\mathsf{RCP-FE}}_{\mathcal{A}}(\lambda)$ は下記のように計算できる．

$$\begin{aligned}
\mathsf{Adv}^{\mathsf{RCP-FE}}_{\mathcal{A}}(\lambda) &= \mathsf{Adv}^{(0)}_{\mathcal{A}}(\lambda) \\
&\leq \left|\mathsf{Adv}^{(0)}_{\mathcal{A}}(\lambda) - \mathsf{Adv}^{(1)}_{\mathcal{A}}(\lambda)\right| \\
&\quad + \sum_{h=0}^{\nu-1} \left|\mathsf{Adv}^{(2-h)}_{\mathcal{A}}(\lambda) - \mathsf{Adv}^{(2-h^+)}_{\mathcal{A}}(\lambda)\right| \\
&\quad + \sum_{h=0}^{\nu-1} \left|\mathsf{Adv}^{(2-h^+)}_{\mathcal{A}}(\lambda) - \mathsf{Adv}^{(2-(h+1))}_{\mathcal{A}}(\lambda)\right| \\
&\quad + \left|\mathsf{Adv}^{(2-\nu)}_{\mathcal{A}}(\lambda) - \mathsf{Adv}^{(3)}_{\mathcal{A}}(\lambda)\right| \\
&\quad + \left|\mathsf{Adv}^{(3)}_{\mathcal{A}}(\lambda)\right| \\
&\leq \mathsf{Adv}^{\mathsf{P1}}_{\mathcal{B}_1}(\lambda) + \sum_{h=0}^{\nu-1} \mathsf{Adv}^{\mathsf{P2}}_{\mathcal{B}^+_{2,h}}(\lambda) \\
&\quad + \sum_{h=0}^{\nu-1} \mathsf{Adv}^{\mathsf{P2}}_{\mathcal{B}_{2,h+1}}(\lambda) + (2d\nu + 6\nu + d + 2)/q \\
&\leq \mathsf{Adv}^{\mathsf{DLIN}}_{\mathcal{E}_1}(\lambda) \\
&\quad + \sum_{h=0}^{\nu-1} \left(\mathsf{Adv}^{\mathsf{DLIN}}_{\mathcal{E}^+_{2,h}}(\lambda) + \mathsf{Adv}^{\mathsf{DLIN}}_{\mathcal{E}_{2,h+1}}(\lambda)\right) \\
&\quad + (2d\nu + 16\nu + 2d + 8)/q
\end{aligned}$$

以上により，Theorem 2 が成り立つ． □

次に，上記証明で利用した Lemma を下記に示す．

Lemma 9. 任意の確率的多項式時間攻撃者 $\mathcal{A}$ に対して，$|\mathsf{Adv}^{(0)}_{\mathcal{A}}(\lambda) - \mathsf{Adv}^{(1)}_{\mathcal{A}}(\lambda)| \leq \mathsf{Adv}^{\mathsf{P1}}_{\mathcal{B}_1}(\lambda) + (d+1)/q$ となる計算量仮定 *Problem 1* に対する確率的多項式時間攻撃者 $\mathcal{B}_1$ が存在する．

Lemma 10. 任意の確率的多項式時間攻撃者 $\mathcal{A}$ に対して，$|\mathsf{Adv}^{(2-h)}_{\mathcal{A}}(\lambda) - \mathsf{Adv}^{(2-h^+)}_{\mathcal{A}}(\lambda)| \leq \mathsf{Adv}^{\mathsf{P2}}_{\mathcal{B}^+_{2,h}}(\lambda) + (d+3)/q$ となる計算量仮定 *Problem 2* に対する確率的多項式時間攻撃者 $\mathcal{B}^+_2$ が存在する．

Lemma 11. 任意の確率的多項式時間攻撃者 $\mathcal{A}$ に対して，

Game0: オリジナルのゲーム．チャレンジ暗号文は，正規の手順で生成した基底を用いて，下記のように正規の暗号文を生成する．

$\boldsymbol{c}_0 := (-s_0, 0, \zeta, 0, \eta_0)_{\mathbb{B}_0}$,
for $i = 1, \ldots, \ell$;
if $\rho(i) = (t, \overrightarrow{v}_i)$, $\boldsymbol{c}_i := (s_i \overrightarrow{e}_{t,1} + \theta_i \overrightarrow{v}_i, 0^{n_t}, 0^{n_t}, \eta_i)_{\mathbb{B}_t}$,
if $\rho(i) = \neg(t, \overrightarrow{v}_i)$, $\boldsymbol{c}_i := (s_i \overrightarrow{v}_i, 0^{n_t}, 0^{n_t}, \eta_i)_{\mathbb{B}_t}$,
$c_{d+1} := g_T^{\zeta} m$,

同様に，プロキシー鍵とユーザ秘密鍵も正規の手順で生成した基底を用いて正規の手順で生成する．

$\boldsymbol{k}_0^* := (\delta, 0, 1, \varphi_0, 0)_{\mathbb{B}_0^*}$,
$\boldsymbol{k}_t^* := (\delta \overrightarrow{x}_t, 0^{n_t}, \overrightarrow{\varphi}_t, 0)_{\mathbb{B}_t^*}$ for $(t, \overrightarrow{x}_t) \in \Gamma$,
$\mathsf{pxk}_{\Gamma,KID} := (\Gamma, \{\boldsymbol{k}_t^*\}_{(t,\overrightarrow{x}_t)\in\Gamma})$,
$\mathsf{sk}_{KID} := (\boldsymbol{k}_0^*)$,

Game1: チャレンジ暗号文を下記の様に生成する点だけが Game0 と異なる．

$w_0 \xleftarrow{\cup} \mathbb{F}_q, \overrightarrow{w}_i, \overrightarrow{\overrightarrow{w}}_i \xleftarrow{\cup} \mathbb{F}_q^{n_t}$ for $i = 1, \ldots, \ell$,
$\boldsymbol{c}_0 := (-s_0, \boxed{w_0}, \zeta, 0, \eta_0)_{\mathbb{B}_0}$,
for $i = 1, \ldots, \ell$;
if $\rho(i) = (t, \overrightarrow{v}_i)$, $\boldsymbol{c}_i := (s_i \overrightarrow{e}_{t,1} + \theta_i \overrightarrow{v}_i, \boxed{\overrightarrow{w}_i}, 0^{n_t}, \eta_i)_{\mathbb{B}_t}$,
if $\rho(i) = \neg(t, \overrightarrow{v}_i)$, $\boldsymbol{c}_i := (s_i \overrightarrow{v}_i, \boxed{\overrightarrow{\overrightarrow{w}}_i}, 0^{n_t}, \eta_i)_{\mathbb{B}_t}$,

Game2-h^+ ($h = 0, \cdots, \nu - 1$): プロキシー鍵，ユーザ秘密鍵，チャレンジ暗号文の下記の要素の生成方法だけが Game2-h と異なる．なお，Game2-0 は Game1 と同一である．

$r_0 \xleftarrow{\cup} \mathbb{F}_q, \overrightarrow{g} \xleftarrow{\cup} \mathbb{F}_q^r, \overrightarrow{a}^T := (a_1, \ldots, a_\ell)^T := M \cdot \overrightarrow{g}^T, \pi_i \xleftarrow{\cup} \mathbb{F}_q (i = 1, \ldots, \ell)$,
$Z_t \xleftarrow{\cup} GL(n_t, \mathbb{F}_q), U_t := (Z_t^{-1})^T$ for $t = 1, \ldots, d$,
$\boldsymbol{k}_0^* := (\delta, \boxed{r_0}, 1, \varphi_0, 0)_{\mathbb{B}_0^*}$,
$\boldsymbol{k}_t^* := (\delta \overrightarrow{x}_t, \boxed{\overrightarrow{x}_t \cdot U_t}, \overrightarrow{\varphi}_t, 0)_{\mathbb{B}_t^*}$ for $(t, \overrightarrow{x}_t) \in \Gamma$,
for $i = 1, \ldots, \ell$;
if $\rho(i) = (t, \overrightarrow{v}_i)$, $\boldsymbol{c}_i := (s_i \overrightarrow{e}_{t,1} + \theta_i \overrightarrow{v}_i, \boxed{(a_i \overrightarrow{e}_{t,1} + \pi_i \overrightarrow{v}_i) \cdot Z_t}, 0^{n_t}, \eta_i)_{\mathbb{B}_t}$,
if $\rho(i) = \neg(t, \overrightarrow{v}_i)$, $\boldsymbol{c}_i := (s_i \overrightarrow{v}_i, \boxed{a_i \overrightarrow{v}_i \cdot Z_t}, 0^{n_t}, \eta_i)_{\mathbb{B}_t}$,

Game2-$(h+1)$ ($h = 0, \cdots, \nu - 1$): プロキシー鍵とチャレンジ暗号文の下記の要素の生成方法だけが Game2-h^+ と異なる．

$\overrightarrow{w}_i, \overrightarrow{\overrightarrow{w}}_i \xleftarrow{\cup} \mathbb{F}_q^{n_t}$ for $i = 1, \ldots, \ell$,
$\boldsymbol{k}_t^* := (\delta \overrightarrow{x}_t, \boxed{0^{n_t}}, \overrightarrow{\varphi}_t, 0)_{\mathbb{B}_t^*}$ for $(t, \overrightarrow{x}_t) \in \Gamma$,
for $i = 1, \ldots, \ell$;
if $\rho(i) = (t, \overrightarrow{v}_i)$, $\boldsymbol{c}_i := (s_i \overrightarrow{e}_{t,1} + \theta_i \overrightarrow{v}_i, \boxed{\overrightarrow{w}_i}, 0^{n_t}, \eta_i)_{\mathbb{B}_t}$,
if $\rho(i) = \neg(t, \overrightarrow{v}_i)$, $\boldsymbol{c}_i := (s_i \overrightarrow{v}_i, \boxed{\overrightarrow{\overrightarrow{w}}_i}, 0^{n_t}, \eta_i)_{\mathbb{B}_t}$,

Game3: チャレンジ暗号文の下記の要素の生成方法だけが Game2-ν と異なる．

$\zeta' \xleftarrow{\cup} \mathbb{F}_q$,
$\boldsymbol{c}_0 := (-s_0, \omega_0, \boxed{\zeta'}, 0, \eta_0)_{\mathbb{B}_0}$,
$c_{d+1} := g_T^{\zeta} m^{(b)}$,

図 **A·1** 失効機能対応暗号文ポリシ型関数型暗号の安全性証明のゲーム列
Fig. A·1 Sequence of games for the revocable ciphertext-policy functional encryption.

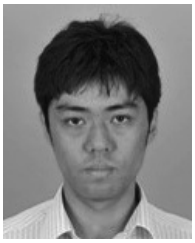
$|\mathsf{Adv}_{\mathcal{A}}^{(2-h^+)}(\lambda) - \mathsf{Adv}_{\mathcal{A}}^{(2-(h+1))}(\lambda)| \leq \mathsf{Adv}_{\mathcal{B}_{2,h+1}}^{\mathsf{P2}}(\lambda) + (d+3)/q$ となる計算量仮定 *Problem 2* に対する確率的多項式時間攻撃者 $\mathcal{B}_2$ が存在する．

Lemma 12. 任意の確率的多項式時間攻撃者 $\mathcal{A}$ に対して，$\mathsf{Adv}_{\mathcal{A}}^{(3)}(\lambda) \leq \mathsf{Adv}_{\mathcal{A}}^{(2-\nu)}(\lambda) + 1/q$ である．

Lemma 13. 任意の確率的多項式時間攻撃者 $\mathcal{A}$ に対して，$\mathsf{Adv}_{\mathcal{A}}^{(3)}(\lambda) = 0$ である．

松田 規

1995 年中央大学理工学部電気・電子工学科卒業．1997 年同大学大学院修士課程修了．同年三菱電機株式会社情報技術総合研究所入社．2022 年静岡大学創造科学技術大学院博士課程修了．博士（工学）．現在，情報セキュリティ技術に関する研究開発に従事．

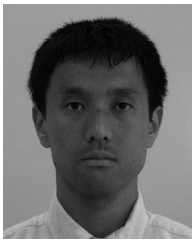


川合 豊

2007 年電気通信大学電気通信学部情報通信工学科卒業．2009 年同大学大学院修士課程修了．2012 年東京大学大学院新領域創成科学研究科複雑理工学研究科博士課程修了．同年三菱電機株式会社情報技術総合研究所入社．博士（工学）．現在，情報セキュリティ技術に関する研究開発に従事．

平野 貴人 （正会員）

2004 年大阪教育大学教育学部教養学科数理科学専攻卒業．2006 年東京都立大学大学院理学研究科数学専攻修士課程修了．2010 年東京工業大学大学院情報理工学研究科数理・計算科学専攻博士課程修了．同年三菱電機株式会社情報技術総合研究所入社．博士（理学）．現在，情報セキュリティ技術に関する研究開発に従事．



伊藤 隆

2000 年東京大学工学部計数工学科卒業．2002 年同大学大学院修士課程修了．同年三菱電機株式会社情報技術総合研究所入社．情報セキュリティ技術に関する研究開発に従事．

西垣 正勝 （正会員）

1990 年静岡大学工学部光電機械工学科卒業．1995 年同大学大学院博士課程修了．日本学術振興会特別研究員（PD）を経て，1996 年静岡大学情報学部助手．同講師，助教授の後，2010 年より同創造科学技術大学院教授．博士（工学）．情報セキュリティ全般，特にヒューマニクスセキュリティ，メディアセキュリティ，ネットワークセキュリティ等に関する研究に従事．2013～2014 年情報処理学会コンピュータセキュリティ研究会主査，2019～2020 年情報環境領域委員長，2020 年調査研究運営委員長．2015～2016 年電子情報通信学会バイオメトリクス研究専門委員会委員長．2016～2020 年日本セキュリティマネジメント学会編集部会長，2021 年より副会長．本会フェロー．