Computer Security Symposium 2016
11 - 13 October 2016

素数位数群における効率的な鍵失効機能付きIDベース暗号の構成法

渡邉 洋平[1,2,a)]　江村 恵太[3,b)]

概要：ID ベース暗号 (Identity-based encryption, IBE) は任意の文字列を公開鍵として用いることのできる公開鍵暗号の一種である．効率的な鍵失効を行う機構は実用上必須であると言えるが，単純な鍵失効では全ユーザ数に対して線形サイズの更新情報を生成する必要がある．この問題を解決するため，全ユーザ数に対して対数サイズの更新情報を用いて効率的に鍵失効が可能である鍵失効機能付き IBE (Revocable IBE, RIBE) が提案されている．本稿では，既存の素数位数群における RIBE の中で最も効率的な方式，具体的には公開パラメータが全ユーザ数に対して定数倍長な方式を提案する．

キーワード：ID ベース暗号，鍵失効機能付き ID ベース暗号，素数位数群，非対称ペアリング．

Efficient Revocable Identity-based Encryption in Prime-order Groups

Yohei Watanabe[1,2,a)]　Keita Emura[3,b)]

Abstract: Identity-based encryption (IBE), which is a type of public-key encryption, allows us to use arbitrary strings as public keys. In particular, an efficient key-revocation procedure in IBE is crucial in terms of practicality, however, the key-update size in a naive solution is linear in the number of users. To provide an efficient procedure with logarithmic sizes of key update, revocable IBE (RIBE) schemes have been investigated so far. In this paper, we propose a new RIBE scheme with constant-size public parameters. The proposed scheme is adaptively secure under static assumptions in prime-order bilinear groups.

Keywords: Identity-based encryption, revocable identity-based encryption, prime-order groups, asymmetric parings.

1. はじめに

Boneh と Franklin [2] により双線型写像を用いた ID ベース暗号 (Identity-based Encryption, IBE) が提案された．IBE では，信頼する鍵生成センタ (Key Generation Center, KGC) が自身のマスター鍵を用いて各ユーザの ID に対する秘密鍵を生成する．また，期間 T ごとに非失効ユーザにのみ ID と T を合わせて $ID\|T$ を新たな ID とみなして秘密鍵を発行することで，期間 T における秘密鍵を持たないユーザをシステムから削除することができる．しかしながら各期間ごとに KGC が非削除ユーザ全員の秘密鍵を発行する必要があるため，スケーラビリティの面で問題がある．この問題を解決するため，Boldyreva ら [1] は KGC の計算コストが各期間ごとに $O(r\log(N/r))$ (N: 最大ユーザ数) である鍵失効機能付き ID ベース暗号 (Revocable IBE, RIBE) を提案した．放送暗号 (Broadcast Encryption) フレームワークの一つである Complete Subtree (CS) 法 [12] を利用することで，スケーラビリティを実現している．

Boldyreva ら以降，様々な RIBE 方式が提案されてきた．適応的安全な方式 [10]，復号鍵漏洩を考慮した方式 [17]，アキュムレータを利用して鍵更新用情報サイズが定数な方式 [20]，属性ベース暗号を利用した方式 [11]，匿名性

[1] 電気通信大学, The University of Electro-Communications. 日本学術振興会特別研究員 PD.
[2] 産業技術総合研究所, AIST.
[3] 情報通信研究機構, NICT.
[a)] watanabe@uec.ac.jp
[b)] k-emura@nict.go.jp

を持つ方式 [3]，格子ベースの方式 [4], [5]，Subset Difference (SD) 法を利用した方式 [5], [9]，階層的 RIBE 方式 [6], [15], [16], [18], [19] などが挙げられる．ここで RIBE として望ましい安全性と効率を考慮すると，適応的安全かつ復号鍵漏洩耐性を持ち，素数位数双線型群で構成されていることが望ましい．しかしながら，これまでこの条件を満足するのは Seo-Emura 方式 [17] のみである[*1]．ただし Seo-Emura 方式は Waters IBE [21] を利用しているため，公開パラメータが最大ユーザ数に対して対数倍長であるという欠点がある．

本稿では，Jutla-Roy IBE [8], [14] を利用した，既存の素数位数群における RIBE の中で最も効率的な方式，具体的には公開パラメータが最大ユーザ数に対して定数倍長な方式を提案する．

2. 準備

まず記法を定義する．“確率的多項式時間” を PPT と省略して書く．素数 p に対して，$\mathbb{Z}_p := \{0, 1, \ldots, p-1\}$，また $\mathbb{Z}_p^\times := \mathbb{Z}_p \setminus \{0\}$ とする． 集合 $\mathcal{X}$ から一様ランダムに要素 x を選ぶことを $x \xleftarrow{\$} \mathcal{X}$ と書く．$\epsilon(\lambda)$ を λ に関する negligible function とする．本稿では一貫して，λ をセキュリティパラメータとして用い，$\mathcal{M}$，$\mathcal{I}$ をそれぞれ（λ によって決まる）平文，ID の集合として用いる．

2.1 双線形写像

双線形写像生成器 $\mathcal{G}$ を，λ を入力し，$(p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, g_1, g_2, e)$ を出力する多項式時間アルゴリズムとする．ここで，p は素数，$\mathbb{G}_1$，$\mathbb{G}_2$，$\mathbb{G}_T$ を位数 p の巡回群，g_1 と g_2 をそれぞれ $\mathbb{G}_1$ と $\mathbb{G}_2$ の生成元とし，e は効率的に計算可能な双線形写像 $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ とする．e は以下の性質を持つ: 任意の $u, u' \in \mathbb{G}_1$ 及び $v, v' \in \mathbb{G}_2$ に対し，$e(uu', v) = e(u, v)e(u', v)$ 及び $e(u, vv') = e(u, v)e(u, v')$ が成り立つ．本稿では，非対称ペアリング，すなわち $\mathbb{G}_1 \neq \mathbb{G}_2$ であるものを考え，また $\mathbb{G}_1$ と $\mathbb{G}_2$ の間に効率的に計算可能な同型写像が知られていないものとする．

2.2 計算量仮定

$\mathcal{A}$ を PPT 攻撃者とし，$\mathcal{A}$ の DDHi 問題（i=1,2）に対するアドバンテージを以下のように定義する．

$$Adv_{\mathcal{G},\mathcal{A}}^{DDHi}(\lambda) := \left| \Pr\left[b' = b \;\middle|\; \begin{array}{l} D := (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, g_1, g_2, e) \leftarrow \mathcal{G}(\lambda), \\ c_1, c_2 \xleftarrow{\$} \mathbb{Z}_p,\ b \xleftarrow{\$} \{0,1\}, \\ \text{if } b = 0 \text{ then } Z := g_i^{c_1 c_2},\ \text{else } Z \xleftarrow{\$} \mathbb{G}_i, \\ b' \leftarrow \mathcal{A}(\lambda, D, g_i^{c_1}, g_i^{c_2}, Z) \end{array} \right] - \frac{1}{2} \right|.$$

[*1] なお，適応的安全かつ復号鍵漏洩耐性を持ち，素数位数群上で構成され，かつ CCA 安全な方式も提案されている [7] が，依然公開パラメータが ID の長さに依存する．[7] のアイディアを提案方式に用いることで，公開パラメータの長さが ID の長さに依存しない CCA 安全な方式が構成できる．

定義 1 (DDHi 仮定)．全ての *PPT* 攻撃者 $\mathcal{A}$ に対して $Adv_{\mathcal{G},\mathcal{A}}^{DDHi}(\lambda) < \epsilon(\lambda)$ ならば，（$\mathcal{G}$ に関する）*DDHi* 仮定が成り立つという．

更に本稿では，DDH1 仮定を基にした新たな仮定として，*Augmented DDH1 (ADDH1)* 仮定を導入する．この仮定は，DDH2 仮定を自然に拡張した DDH2v 仮定 [13] と似たものであり，それほど人工的な仮定ではない．紙面の都合上，本仮定の妥当性の証明は割愛する．PPT 攻撃者 $\mathcal{A}$ の ADDH1 問題に対するアドバンテージは，上記 DDH1 問題において $\mathcal{A}$ に与えるインスタンスに，$g_1^{dc_3}, g_2^{d}, g_2^{c_2 c_3}, g_2^{dc_3}, g_2^{\frac{1}{c_3}}$ $(d \xleftarrow{\$} \mathbb{Z}_p, c_3 \xleftarrow{\$} \mathbb{Z}_p^\times)$ を加えることで定義される．

定義 2 (ADDH1 仮定)．全ての *PPT* 攻撃者 $\mathcal{A}$ に対して $Adv_{\mathcal{G},\mathcal{A}}^{ADDH1}(\lambda) < \epsilon(\lambda)$ ならば，（$\mathcal{G}$ に関する）*ADDH1* 仮定が成り立つという．

2.3 KUNode アルゴリズム

既存研究 [1], [10], [17] 同様，以下の KUNode アルゴリズムを用いる．KUNode アルゴリズムは，二分木 BT，削除リスト RL，期間 $T \in \mathcal{T}$ を入力し，ノードの集合を出力する．η が葉ノードではない時，η_L と η_R をそれぞれその左側，右側の子ノードとする．η が葉ノードの時，$\mathsf{Path}(\mathtt{BT}, \eta)$ は η からルートまでの経路上のノードの集合とする．各ユーザは葉ノードに割り当てられ，もし η に割り当てられたユーザの鍵を期間 $T \in \mathcal{T}$ に失効する場合，$(\eta, T) \in RL$ となる．$\mathsf{KUNode}(\mathtt{BT}, RL, T)$ は次のように実行される．$\mathcal{X} := \emptyset$，$\mathcal{Y} := \emptyset$ とする．任意の $(\eta_i, T_i) \in RL$ に対して，$T_i \leq T$ ならば $\mathsf{Path}(\mathtt{BT}, \eta_i)$ を $\mathcal{X}$ に加える．また任意の $\eta \in \mathcal{X}$ に対して，$\eta_L \notin \mathcal{X}$ ならば，η_L を $\mathcal{Y}$ に加える．もし $\eta_R \notin \mathcal{X}$ ならば，η_R を $\mathcal{Y}$ に加える．最終的に $\mathcal{Y} \neq \emptyset$ であれば $\mathcal{Y}$ を出力し，そうでなければルートを $\mathcal{Y}$ に加え，出力する．

2.4 鍵失効機能付き ID ベース暗号

本稿で扱う RIBE のモデル [17], [19] について記述する．RIBE II は次の 7 つのアルゴリズムからなる．以下では Setup 以外の入力からマスター公開鍵の記述を省略する．

- $(mpk, msk, RL, st) \leftarrow \mathsf{Setup}(\lambda, N)$: 確率的アルゴリズムであり，$\lambda$ と最大ユーザ数 N を入力し，マスター公開鍵 mpk，マスター秘密鍵 msk，削除リスト $RL = \emptyset$，及び状態情報 st を出力する．
- $sk_{\mathtt{I}} \leftarrow \mathsf{SKGen}(st, \mathtt{I})$: st と ID $\mathtt{I} \in \mathcal{I}$ を入力し，$\mathtt{I}$ の秘密鍵 $sk_{\mathtt{I}}$ を出力する．
- $ku_T \leftarrow \mathsf{KeyUp}(msk, st, RL, T)$: msk，st，RL，期間 $T \in \mathcal{T}$ を入力し，T における更新情報 ku_T を出力する．
- $dk_{\mathtt{I},T}$ or $\perp \leftarrow \mathsf{DKGen}(sk_{\mathtt{I}}, ku_T)$: 確率的アルゴリズムであり，$sk_{\mathtt{I}}$ と ku_T を入力し，復号鍵 $dk_{\mathtt{I},T}$ を，または $(\mathtt{I}, T) \in RL$ の場合 $\perp$ を出力する．
- $C_{\mathtt{I},T} \leftarrow \mathsf{Enc}(M, \mathtt{I}, T)$: 確率的アルゴリズムであり，平

文 $M \in \mathcal{M}$，$\mathtt{I} \in \mathcal{I}$，$T \in \mathcal{T}$ を入力し，暗号文 $C_{\mathrm{I},T}$ を出力する．

– M or $\perp \leftarrow \mathsf{Dec}(dk_{\mathrm{I},T}, C_{\mathrm{I},T})$: 確定的アルゴリズムであり，$dk_{\mathrm{I},T}$ と $C_{\mathrm{I},T}$ を入力し，M または $\perp$ を出力する．

– $RL \leftarrow \mathsf{Revoke}(\mathtt{I}, T, RL, st)$: $(\mathtt{I}, T) \in \mathcal{I} \times \mathcal{T}$，現在の削除リスト RL，st を入力し，RL を更新し，出力する．

上記のモデルでは，以下の正当性を満たすものとする: 全ての $\lambda \in \mathbb{N}$，全ての $(mpk, msk, RL, st) \leftarrow \mathsf{Setup}(\lambda, N)$，全ての $M \in \mathcal{M}$，全ての $\mathtt{I} \in \mathcal{I}$，全ての $T \in \mathcal{T}$ に対して，$(\mathtt{I}, T) \notin RL$ であるならば，$M = \mathsf{Dec}(\mathsf{DKGen}(\mathsf{SKGen}(st, \mathtt{I}), \mathsf{KeyUp}(msk, st, RL, T)), \mathsf{Enc}(M, \mathtt{I}, T))$ が成り立つ．

次に，RIBE の安全性として選択平文攻撃に対する識別不可能性（IND-RID-CPA）を定義する．本稿で考える安全性は，復号鍵漏洩耐性 [17] も考慮した定義である．$\mathcal{A}$ を PPT 攻撃者とし，IND-RID-CPA ゲームにおける $\mathcal{A}$ のアドバンテージを以下のように定義する．

$$Adv^{IND\text{-}RID\text{-}CPA}_{\Pi,\mathcal{A}}(\lambda, N) := \left| \Pr\left[b' = b \;\middle|\; \begin{array}{l} (mpk, msk, RL, st) \leftarrow \mathsf{Setup}(\lambda, N), \\ (M_0^*, M_1^*, \mathtt{I}^*, T^*, state) \\ \qquad \leftarrow \mathcal{A}^{\mathcal{O}}(\mathsf{find}, mpk), \\ b \xleftarrow{\$} \{0,1\}, C^*_{\mathtt{I}^*,T^*} \leftarrow \mathsf{Enc}(M_b^*, \mathtt{I}^*, T^*), \\ b' \leftarrow \mathcal{A}^{\mathcal{O}}(\mathsf{guess}, C^*_{\mathtt{I}^*,T^*}, state) \end{array} \right] - \frac{1}{2} \right|.$$

ここで，$\mathcal{O}$ は 4 つのオラクル $SKGen(\cdot)$，$KeyUp(\cdot)$，$Revoke(\cdot,\cdot)$，$DKGen(\cdot,\cdot)$ を表し，具体的にはそれぞれ以下のように定義される．

SKGen$(\cdot)$: クエリ $\mathtt{I} \in \mathcal{I}$ に対し，$\mathsf{SKGen}(st, \mathtt{I})$ を返す．

KeyUp$(\cdot,\cdot)$: クエリ $T \in \mathcal{T}$ に対し，$\mathsf{KeyUp}(msk, st, RL, T)$ を返す．

Revoke$(\cdot,\cdot)$: クエリ $(\mathtt{I}, T) \in \mathcal{I} \times \mathcal{T}$ に対し，$\mathsf{Revoke}(\mathtt{I}, T, RL, st)$ を実行する．

DKGen$(\cdot,\cdot)$: クエリ $(\mathtt{I}, T) \in \mathcal{I} \times \mathcal{T}$ に対し，$\mathsf{DKGen}(sk_{\mathrm{I}}, ku_T)$ を返す．

$\mathcal{A}$ は以下の制約を除き，上記オラクルに自由にアクセス可能である: (1) クエリ $KeyUp(T)$ 及び $Revoke(\cdot, T)$ は，全ての過去のクエリにおける期間 T' に対して $T \geq T'$ を満たさなければならない．(2) $KeyUp(T)$ をクエリした以降は $Revoke(\cdot, T)$ をクエリできない．(3) $SKGen(\mathtt{I}^*)$ をクエリするならば，必ず $\mathsf{Revoke}(\mathtt{I}^*, T\ (\leq T^*))$ をクエリしなければならない．(4) $DKGen(\cdot, T)$ は $KeyUp(T)$ をクエリした以降にしかクエリできない．(5) $DKGen(\mathtt{I}^*, T^*)$ はクエリできない．

定義 3 (IND-RID-CPA)．全ての *PPT* 攻撃者 $\mathcal{A}$ に対して，$Adv^{IND\text{-}RID\text{-}CPA}_{\Pi,\mathcal{A}}(\lambda, N) < \epsilon(\lambda)$ ならば，*RIBE* Π は *IND-RID-CPA* 安全であるという．

3. Jutla-Roy IBE の拡張

1 節で述べたように，本稿では提案する構成法の安全性を，基とする Jutla-Roy IBE の安全性に帰着する形で証明する．そのため本節では，そのような帰着が可能になるよう Jutla-Roy IBE [8], [14] を拡張し，これを便宜的に拡張 JR-IBE と呼ぶ．具体的には，公開パラメータに 4 つの群要素 $(\chi_1, g_2^{x_0\beta}, g_2^{y_0\beta}, g_2^{\frac{1}{\beta}})$ を追加する．これらの要素は方式内で使われることはないが，これらを公開してもなお安全であることを証明するためには証明に工夫が必要となる．

3.1 構成法

拡張 JR-IBE Π_{JR} は以下のように構成される[*2]．

- $\mathsf{Init}(\lambda)$: $(\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, p, g_1, g_2, e) \leftarrow \mathcal{G}$ を実行する．$x_0, y_0, x_1, y_1, x_2, y_2, x_3, y_3 \xleftarrow{\$} \mathbb{Z}_p$，$\alpha, \beta \xleftarrow{\$} \mathbb{Z}_p^\times$ を選び，以下を計算する．

$$z = e(g_1, g_2)^{-x_0\alpha + y_0},\ u_1 := g_1^{-x_1\alpha + y_1},\ w_1 := g_1^{-x_2\alpha + y_2},\ h_1 := g_1^{-x_3\alpha + y_3},\ \chi_1 := g_1^{\beta(-x_0\alpha + y_0)}.$$

公開パラメータ $PP := (g_1, g_1^\alpha, u_1, w_1, h_1, \chi_1, g_2, g_2^{x_1}, g_2^{y_1}, g_2^{x_2}, g_2^{x_3}, g_2^{y_3}, z, g_2^{x_0\beta}, g_2^{y_0\beta}, g_2^{\frac{1}{\beta}})$，マスター秘密鍵 $MK := (g_2^{y_0}, g_2^{-x_0})$ を出力する．

- $\mathsf{KeyGen}(mk, \mathtt{I})$: $MK = (d_1', d_2')$ とする．$r \xleftarrow{\$} \mathbb{Z}_p$ を選び，以下を計算し，$SK_{\mathrm{I}} := (D_1, D_1', D_2, D_2', D_3)$ として出力する．

$$D_1 := (g_2^{y_2})^r,\ \ D_1' := d_1' \left((g_2^{y_1})^{\mathtt{I}} g_2^{y_3}\right)^r,$$
$$D_2 := (g_2^{x_2})^{-r},\ \ D_2' := d_2' \left((g_2^{x_1})^{\mathtt{I}} g_2^{x_3}\right)^{-r},\ \ D_3 := g_2^r.$$

- $\mathsf{IBEnc}(\mathtt{I}, M)$: $t, \mathtt{tag} \xleftarrow{\$} \mathbb{Z}_p$ を選び，$M \in \mathbb{G}_T$ に対して以下を計算し，$C := (C_0, C_1, C_2, C_3, \mathtt{tag})$ を出力する．

$$C_0 := Mz^t,\ C_1 := g_1^t,\ C_2 := (g_1^\alpha)^t,\ C_3 := \left(u_1^{\mathtt{I}} w_1^{\mathtt{tag}} h_1\right)^t.$$

- $\mathsf{IBDec}(SK_{\mathrm{I}}, C)$: $SK_{\mathrm{I}} = (D_1, D_1', D_2, D_2', D_3)$ and $C = (C_0, C_1, C_2, C_3, \mathtt{tag})$ とし，以下を計算する．

$$M = \frac{C_0 e(C_3, D_3)}{e(C_1, D_1^{\mathtt{tag}} D_1') e(C_2, D_2^{\mathtt{tag}} D_2')}.$$

上記構成法では，正しく生成された暗号文は正しく生成された秘密鍵を用いれば必ず元の平文に復号可能だが，紙面の都合上，詳細は割愛する．

3.2 拡張 JR-IBE の安全性

定理 1. *ADDH1* 仮定及び *DDH2* 仮定が成り立つならば，上記拡張 *JR-IBE* Π_{JR} は *IND-ID-CPA* 安全である．

証明. **Semi-functional 暗号文:** 通常の暗号文 C を $(C_0, C_1, C_2, C_3, \mathtt{tag})$ とする．$\mu \xleftarrow{\$} \mathbb{Z}_p$ に対して，semi-functional 暗号文 $\widetilde{C} := (\tilde{C}_0, \tilde{C}_1, \tilde{C}_2, \tilde{C}_3, \widetilde{\mathtt{tag}})$ を次のように計算する: $\widetilde{\mathtt{tag}} := \mathtt{tag}$,

[*2] 紙面の都合上，IBE のモデルは割愛する．

$$\tilde{C}_0 := C_0 e(g_1, g_2)^{-x_0\mu},\ \tilde{C}_1 := C_1,\ \tilde{C}_2 := C_2 g_1^{\mu},$$
$$\tilde{C}_3 := C_3 \Big((g_1^{x_1})^{\mathtt{I}} (g_1^{x_2})^{\mathtt{tag}} g_1^{x_3} \Big)^{-\mu}.$$

Semi-functional 秘密鍵: 通常の秘密鍵 $sk_{\mathtt{I}}$ を $(D_1, D_1', D_2, D_2', D_3)$ とする．$\gamma, \phi \xleftarrow{\$} \mathbb{Z}_p$ に対して，semi-functional 秘密鍵 $\tilde{sk}_{\mathtt{I}} := (\tilde{D}_1, \tilde{D}_1', \tilde{D}_2, \tilde{D}_2', \tilde{D}_3)$ を次のように計算する:

$$\tilde{D}_1 := D_1 g_2^{\gamma},\ \ \tilde{D}_1' := D_1' g_2^{\gamma\phi},$$
$$\tilde{D}_2 := D_2 g_2^{-\frac{\gamma}{\alpha}},\ \ \tilde{D}_2' := D_2' g_2^{-\frac{\gamma\phi}{\alpha}},\ \ \tilde{D}_3 := D_3.$$

$\mathtt{I}$ に対する semi-functional 暗号文が $\mathtt{I}$ の（通常の）秘密鍵で復号可能なことは，$(e(g_1, g_2)^{-x_0\mu}. e(g_1^{-\mu(x_1\mathtt{I}+x_2\mathtt{tag}+x_3)}, D_3))/e(g_1^{\mu}, D_2^{\mathtt{tag}} D_2') = 1_{\mathbb{G}_T}$ から分かる．ここで，$1_{\mathbb{G}_T}$ は $\mathbb{G}_T$ の単位元を表す．また，通常の暗号文も semi-functional 秘密鍵で復号可能なことは，$e(C_1, g_2^{\gamma\mathtt{tag}} g_2^{\gamma\phi}) e(C_2, g_2^{-\frac{\gamma}{\alpha}\mathtt{tag}} g_2^{-\frac{\gamma\phi}{\alpha}}) = 1_{\mathbb{G}_T}$ から確認できる．

以下のゲーム列を定義する．

$\mathsf{Game}_{\mathsf{Real}}$: 通常の IND-ID-CPA ゲーム．

Game_0: チャレンジ暗号文が semi-functional であること以外は $\mathsf{Game}_{\mathsf{Real}}$ と同じゲーム．

Game_k $(1 \le k \le q)$: 以下の点を除き，Game_0 と同じゲーム: q を秘密鍵生成オラクルである $KeyGen$ オラクルへの最大クエリ数とし，$\mathtt{I}_i$ $(1 \le i \le q)$ を i 番目のクエリとする．$KeyGen$ オラクルは，最初の k 個のクエリ $\mathtt{I}_1, \ldots, \mathtt{I}_k$ に関して semi-functional 秘密鍵を返し，残りのクエリ（$\mathtt{I}_{k+1}, \ldots, \mathtt{I}_q$）には通常の秘密鍵を返す．

$\mathsf{Game}_{\mathsf{Final}}$: チャレンジ暗号文が $\mathbb{G}_T$ のランダムな要素の semi-fucntional 暗号文であること以外は Game_q と同じゲーム．

S_{Real}, S_k $(0 \le k \le q)$, and S_{Final} を，それぞれ各ゲーム $\mathsf{Game}_{\mathsf{Real}}$, Game_k, and $\mathsf{Game}_{\mathsf{Final}}$ で $b' = b$ となる確率とすると，$Adv^{IND\text{-}ID\text{-}CPA}_{\Pi_{\mathrm{JR}},\mathcal{A}}(\lambda) \le |S_{\mathsf{Real}} - S_0| + \sum_{i=1}^{q} |S_{i-1} - S_i| + |S_q - S_{\mathsf{Final}}| + |S_{\mathsf{Final}} - \frac{1}{2}|$ である．

補題 1. $|S_{Real} - S_0| \le 2Adv^{DDH1}_{\mathcal{G},\mathcal{B}}(\lambda)$.

Proof. PPT アルゴリズム $\mathcal{B}$ は DDH1 問題のインスタンス $(g_1, g_1^{c_1}, g_1^{c_2}, g_2, Z)$ を得る．$\mathcal{B}$ は $x_0, y_0, x_1, y_1, x_2, y_2, x_3, y_3 \xleftarrow{\$} \mathbb{Z}_p$, $\beta \xleftarrow{\$} \mathbb{Z}_p^{\times}$ を選び，$\alpha := c_1$ とし，以下を計算する．

$$z := e(g_1^{c_1}, g_2)^{-x_0} e(g_1, g_2)^{y_0},\ u_1 := (g_1^{c_1})^{-x_1} g_1^{y_1},$$
$$w_1 := (g_1^{c_1})^{-x_2} g_1^{y_2},\ h_1 := (g_1^{c_1})^{-x_3} g_1^{y_3},\ \chi_1 := (g_1^{c_1})^{-x_0\beta} g_1^{y_0\beta}.$$

$\mathcal{B}$ は mpk を $\mathcal{A}$ に送る．$\mathcal{B}$ は $msk := (g_2^{y_0}, g_2^{-x_0})$ を知っているため，$KeyGen$ オラクルをシミュレートできる．

$\mathcal{B}$ は $\mathcal{A}$ より $(M_0^*, M_1^*, \mathtt{I}^*)$ を受け取り，$d \xleftarrow{\$} \{0,1\}$ を選ぶ．$\mathtt{tag}^* \xleftarrow{\$} \mathbb{Z}_p$ を選び，$t := c_2$ とし，次のように計算した $C^* := (C_0^*, C_1^*, C_2^*, C_3^*, \mathtt{tag}^*)$ を $\mathcal{A}$ に送る．

$$C_0^* := M_d^* e(Z, g_2)^{-x_0} e(g_1^{c_2}, g_2)^{y_0},\ C_1^* := g_1^{c_2},\ C_2^* := Z,$$
$$C_3^* := Z^{-x_1\mathtt{I}^* - x_2\mathtt{tag}^* - x_3} (g_1^{c_2})^{y_1\mathtt{I}^* + y_2\mathtt{tag}^* + y_3}.$$

$b = 0$ のとき，上記暗号文は通常の形になり，$b = 1$ のとき，semi-functional 暗号文となる． □

補題 2. $|S_{k-1} - S_k| \le 2Adv^{DDH2}_{\mathcal{G},\mathcal{B}}(\lambda)$.

Proof. PPT アルゴリズム $\mathcal{B}$ は DDH2 問題のインスタンス $(g_1, g_2, g_2^{c_1}, g_2^{c_2}, Z)$ を得る．$\mathcal{B}$ は $x_0', y_0, x_1', y_1', y_1'', x_2', x_3', y_3', y_3'' \xleftarrow{\$} \mathbb{Z}_p$, $\alpha, \beta \xleftarrow{\$} \mathbb{Z}_p^{\times}$ を選び，$x_0 := \frac{x_0' + y_0}{\alpha}$, $y_1 := y_1' + c_2 y_1''$, $x_1 := \frac{x_1' + y_1}{\alpha}$, $y_2 := c_2$, $x_2 := \frac{x_2' + y_2}{\alpha}$, $y_3 := y_3' + c_2 y_3''$, $x_3 := \frac{x_3' + y_3}{\alpha}$ とし，以下を計算する．

$$z := e(g_1, g_2)^{-x_0'},\ \ u_1 := g_1^{-x_1'},\ \ w_1 := g_1^{-x_2'},\ \ h_1 := g_1^{-x_3'},$$
$$\chi_1 := g_1^{-x_0'\beta},\ \ g_2^{x_1} := g_2^{\frac{x_1' + y_1'}{\alpha}} (g_2^{c_2})^{\frac{y_1''}{\alpha}},\ \ g_2^{y_1} := g_2^{y_1'} (g_2^{c_2})^{y_1''},$$
$$g_2^{x_2} := g_2^{\frac{x_2'}{\alpha}} (g_2^{c_2})^{\frac{1}{\alpha}},\ \ g_2^{y_2} := g_2^{c_2},\ \ g_2^{x_3} := g_2^{\frac{x_3' + y_3'}{\alpha}} (g_2^{c_2})^{\frac{y_3''}{\alpha}},$$
$$g_2^{y_3} := g_2^{y_3'} (g_2^{c_2})^{y_3''}.$$

$\mathcal{B}$ は mpk を $\mathcal{A}$ に送る．ここで，$\mathcal{B}$ は $msk := (g_2^{y_0}, g_2^{-x_0})$ を知っていることに留意されたい．

$\mathcal{B}$ がどのように $KeyGen$ オラクルをシミュレートするかを示す．$\mathtt{I}_i$ $(1 \le i \le q)$ をオラクルへの i 番目のクエリとする．$\mathcal{B}$ は最初の $k-1$ 個は semi-functional 秘密鍵を作り，Z を k 番目の鍵に埋め込み，残りは全て通常の鍵を作る．以下では，$SK_{\mathtt{I}_k} := (D_1, D_1', D_2, D_2', D_3)$ に対して，どのように Z を埋め込むかを示す．

$$D_1 := Z,\ \ D_1' := g_2^{y_0} (g_2^{c_1})^{\mathtt{I}_k y_1' + y_3'} Z^{\mathtt{I}_k y_1'' + y_3''},$$
$$D_2 := \Big((g_2^{c_1})^{x_2'} Z \Big)^{-\frac{1}{\alpha}},$$
$$D_2' := g_2^{-\frac{x_0'}{\alpha}} (g_2^{c_1})^{-\frac{\mathtt{I}_k (x_1' + y_1') + x_3' + y_3'}{\alpha}} g_2^{-\frac{y_0}{\alpha}} Z^{-\frac{\mathtt{I}_k y_1'' + y_3''}{\alpha}},$$
$$D_3 := g_2^{c_1}.$$

$r := c_1$ とすると，$b = 0$ ならば通常の秘密鍵，$b = 1$ ならば semi-functional 秘密鍵である．実際，以下が成り立つ．

$$D_1 := g_2^{c_1 c_2 + \gamma} = g_2^{y_2 r + \gamma},$$
$$D_1' := g_2^{y_0 + c_1(\mathtt{I}_k(y_1' + c_2 y_1'') + y_3' + c_2 y_3'')} g_2^{\gamma(\mathtt{I}_k y_1'' + y_3'')}$$
$$= g_2^{y_0 + r(\mathtt{I}_k y_1 + y_3)} g_2^{\gamma\phi},$$
$$D_2 := g_2^{-\frac{c_1(x_2' + c_2)}{\alpha}} g_2^{-\frac{\gamma}{\alpha}} = g_2^{-r x_2} g_2^{-\frac{\gamma}{\alpha}},$$
$$D_2' := g_2^{-\frac{(x_0' + y_0) + c_1(\mathtt{I}_k(x_1' + y_1' + c_2 y_1'') + (x_3' + y_3' + c_2 y_3''))}{\alpha}} g_2^{-\frac{\gamma(\mathtt{I}_k y_1'' + y_3'')}{\alpha}}$$
$$= g_2^{-x_0 - r(\mathtt{I}_k x_1 + x_3)} g_2^{-\frac{\gamma\phi}{\alpha}},$$

ここで，$Z := g_2^{c_1 c_2 + \gamma}$, $\phi := \mathtt{I}_k y_1'' + y_3''$ である．y_1'' と y_3'' は一様ランダムに選ばれているため，$\mathcal{A}$ から ϕ は一様ランダムに選ばれたように見える．

$\mathcal{B}$ は $\mathcal{A}$ より $(M_0^*, M_1^*, \mathtt{I}^*)$ を受け取ったら，$d \xleftarrow{\$} \{0,1\}$ を選び，M_d^* のチャレンジ semi-functional 暗号文を作るが，そのためには $\mathcal{B}$ は c_2 を知らなければならない．そこで，$\widetilde{\mathtt{tag}}^* := -\mathtt{I}^* y_1'' - y_3''$ とすることでその問題を解決する．ここで，y_1'' と y_3'' は一様ランダムに選ばれ，また $\mathtt{I}_k \neq \mathtt{I}^*$ なため，$\mathcal{A}$ からは $\widetilde{\mathtt{tag}}^*$ が一様ランダムに選ばれたように見える．$\mathcal{B}$ は $t, \mu \xleftarrow{\$} \mathbb{Z}_p$ を選び，次のように $\widetilde{C}^* := (\tilde{C}_0^*, \tilde{C}_1^*, \tilde{C}_2^*, \tilde{C}_3^*, \widetilde{\mathtt{tag}}^*)$ を計算し，$\mathcal{A}$ に送る．

$$
\begin{aligned}
\tilde{C}_0^* :=& M_d^* z^t e(g_1, g_2)^{-x_0 \mu} = M_d^* e(g_1, g_2)^{-x_0(\alpha t + \mu) + y_0 s}, \\
\tilde{C}_1^* :=& g_1^t, \quad \tilde{C}_2^* := g_1^{\alpha t + \mu} \\
\tilde{C}_3^* :=& \left(u_1^{\mathtt{I}^*} w_1^{\widetilde{\mathtt{tag}}^*} h_1\right)^t g_1^{-\frac{\mu}{\alpha}(\mathtt{I}^*(x_1' + y_1') + x_2' \widetilde{\mathtt{tag}}^* + x_3' + y_3')} \\
=& \left(u_1^{\mathtt{I}^*} w_1^{\widetilde{\mathtt{tag}}^*} h_1\right)^t g_1^{-\frac{\mu}{\alpha}(\mathtt{I}^*(x_1' + y_1') + x_2' \widetilde{\mathtt{tag}}^* + x_3' + y_3')} \\
& \qquad \cdot g_1^{-\frac{c_2 \mu}{\alpha}(\mathtt{I}^* y_1'' + \widetilde{\mathtt{tag}}^* + y_3'')} \\
=& \left(u_1^{\mathtt{I}^*} w_1^{\widetilde{\mathtt{tag}}^*} h_1\right)^t g_1^{-\mu(\mathtt{I}^* x_1 + x_2 \widetilde{\mathtt{tag}}^* + x_3)}.
\end{aligned}
$$

□

補題 3. $|S_q - S_{Final}| \leq 2 Adv_{\mathcal{G},\mathcal{B}}^{ADDH1}(\lambda)$.

Proof. PPT アルゴリズム $\mathcal{B}$ は ADDH1 問題のインスタンス $(g_1, g_1^{c_1}, g_1^{c_2}, g_1^{dc_3}, g_2, g_2^d, g_2^{c_2 c_3}, g_2^{dc_3}, g_2^{\frac{1}{c_3}}, Z)$ を得る．$\mathcal{B}$ は $x_1, x_2, x_3, y_1', y_2', y_3' \xleftarrow{\$} \mathbb{Z}_p$，$\alpha \xleftarrow{\$} \mathbb{Z}_p^\times$ を選び，$x_0 := c_2$，$y_0' := d$，$y_0 := x_0 \alpha + y_0'$，$y_1 := x_1 \alpha + y_1'$，$y_2 := x_2 \alpha + y_2'$，$y_3 := x_3 \alpha + y_3'$，$\beta := c_3$，$\beta x_0 := c_2 c_3$，$\beta y_0 := \beta(x_0 \alpha + y_0') = \alpha c_2 c_3 + d c_3$ とし，以下を計算する．

$$
\begin{aligned}
& z := e(g_1, g_2^d) = e(g_1, g_2)^{y_0'}, \quad u_1 := g_1^{y_1'}, \quad w_1 := g_1^{y_2'}, \\
& h_1 := g_1^{y_3'}, \quad \chi_1 := g_1^{dc_3} = g_1^{\beta y_0'}, \\
& g_2^{\beta x_0} := g_2^{c_2 c_3}, \quad g_2^{\beta y_0} := (g_2^{c_2 c_3})^\alpha g_2^{dc_3}, \quad g_2^{\frac{1}{\beta}} := g_2^{\frac{1}{c_3}}.
\end{aligned}
$$

$\mathcal{B}$ は mpk を $\mathcal{A}$ に送る．

KG オラクルへのクエリ $\mathtt{I}$ に対して，$\mathcal{B}$ は $r, \phi', \gamma \xleftarrow{\$} \mathbb{Z}_p$ を選び，次のように $sk_{\mathtt{I}} = (D_1, D_1', D_2, D_2', D_3)$ を計算する．

$$
\begin{aligned}
& D_1 := g_2^{y_2 r + \gamma}, \quad D_1' := g_2^d g_2^{(y_1' \mathtt{I} + y_3') r + \alpha \phi'}, \\
& D_2 := g_2^{-x_2 r - \frac{\gamma}{\alpha}}, \quad D_2' := g_2^{-\phi'}, \quad D_3 := g_2^r.
\end{aligned}
$$

ここで，$\phi' := x_0 + (x_1 \mathtt{I} + x_3) r + \frac{\gamma \phi}{\alpha}$ とおくと，

$$
\begin{aligned}
D_1' &= g_2^{x_0 \alpha + y_0' + ((x_1 \alpha + y_1') \mathtt{I} + x_3 \alpha + y_3') r + \gamma \phi} = g_2^{y_0 + (y_1 \mathtt{I} + y_3) r + \gamma \phi}, \\
D_2' &= g_2^{-x_0 - (x_1 \mathtt{I} + x_3) r - \frac{\gamma \phi}{\alpha}},
\end{aligned}
$$

となり，正しい semi-functional 秘密鍵になっている．

$\mathcal{B}$ は $\mathcal{A}$ から $(M_0^*, M_1^*, \mathtt{I}^*)$ を受け取ったら，$d \xleftarrow{\$} \{0,1\}$ を選ぶ．更に $t, \mathtt{tag}^* \xleftarrow{\$} \mathbb{Z}_p$ を選び，次のように $C^* := (C_0^*, C_1^*, C_2^*, C_3^*, \mathtt{tag}^*)$ を計算する．

$$
\begin{aligned}
& C_0^* := M_d^* \cdot e(g_1, g_2^d)^t e(Z, g_2)^{-1}, \; C_1^* := g_1^t, \; C_2^* := g_1^{\alpha t} g_1^{c_1}, \\
& C_3^* := (u_1^{\mathtt{I}^*} w_1^{\mathtt{tag}^*} h_1)^t (g_1^{c_1})^{-x_1 \mathtt{I}^* - x_2 \mathtt{tag}^* - x_3}.
\end{aligned}
$$

$\mu := c_1$ とすると，もし $b = 0$ ならば M_d^* の semi-functional 暗号文に，$b = 1$ ならば $\mathbb{G}_T$ のランダムな要素の semi-functional 暗号文となる． □

補題 1–3 から，$Adv_{\Pi,\mathcal{A}}^{IND\text{-}ID\text{-}CPA}(\lambda) \leq 2 Adv_{\mathcal{G},\mathcal{B}}^{DDH1}(\lambda) + 2q \cdot Adv_{\mathcal{G},\mathcal{B}}^{DDH2}(\lambda) + 2 Adv_{\mathcal{G},\mathcal{B}}^{ADDH1}(\lambda) \leq 4 Adv_{\mathcal{G},\mathcal{B}}^{ADDH1}(\lambda) + 2q \cdot Adv_{\mathcal{G},\mathcal{B}}^{DDH2}(\lambda)$ が成り立つ． □

4. 提案構成法

Jutla-Roy IBE を基に，Seo, Emura のテクニック [17] を応用し，RIBE を構成する．提案方式の安全性は，前節で示した拡張 JR-IBE 方式の安全性に帰着する．

- $\mathsf{Setup}(\lambda, N)$: $(\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, p, g_1, g_2, e) \leftarrow \mathcal{G}$ を実行する．$x_0, y_0, x_1, y_1, x_2, y_2, x_3, y_3, x_4, y_4, x_5, y_5 \xleftarrow{\$} \mathbb{Z}_p$，$\alpha \xleftarrow{\$} \mathbb{Z}_p^\times$ を選び，以下を計算する．

  $$
  \begin{aligned}
  & z = e(g_1, g_2)^{-x_0 \alpha + y_0}, \; u_1 := g_1^{-x_1 \alpha + y_1}, \; w_1 := g_1^{-x_2 \alpha + y_2}, \\
  & h_1 := g_1^{-x_3 \alpha + y_3}, \; v_1 := g_1^{-x_4 \alpha + y_4}, \; \hat{v}_1 := g_1^{-x_5 \alpha + y_5},
  \end{aligned}
  $$

  $\mathtt{BT}$ を葉ノードが N 個からなる二分木とし，ここでは簡単のため N は 2 のべき乗とする．$mpk := (g_1, g_1^\alpha, u_1, w_1, h_1, v_1, \hat{v}_1, g_2, g_2^{x_1}, g_2^{x_2}, g_2^{x_3}, g_2^{x_4}, g_2^{x_5}, g_2^{y_1}, g_2^{y_2}, g_2^{y_3}, g_2^{y_4}, g_2^{y_5}, z)$，$msk := (g_2^{y_0}, g_2^{-x_0})$，$st := \mathtt{BT}$，$RL := \emptyset$ を出力する．
- $\mathsf{SKGen}(st, \mathtt{I})$: $\mathtt{BT}$ からランダムに葉ノード η を選び，$\mathtt{I}$ を保存する．各ノード $\theta \in \mathsf{Path}(\mathtt{BT}, \eta)$ について，P_θ が定義されていればそれを呼び出す．そうでなければ $P_\theta \xleftarrow{\$} \mathbb{G}_2$ を選び，θ に保存する．$r_\theta \xleftarrow{\$} \mathbb{Z}_p$ を選び，

  $$
  \begin{aligned}
  & \mathtt{SK}_{1,\theta} := (g_2^{y_2})^{r_\theta}, \; \mathtt{SK}_{1,\theta}' := P_\theta \left((g_2^{y_1})^{\mathtt{I}} g_2^{y_3}\right)^{r_\theta}, \\
  & \mathtt{SK}_{2,\theta} := (g_2^{x_2})^{-r_\theta}, \; \mathtt{SK}_{2,\theta}' := P_\theta \left((g_2^{x_1})^{\mathtt{I}} g_2^{x_3}\right)^{-r_\theta}, \\
  & \mathtt{SK}_{3,\theta} := g_2^{r_\theta},
  \end{aligned}
  $$

  を計算し，最終的に $sk_{\mathtt{I}} := \{(\mathtt{SK}_{1,\theta}, \mathtt{SK}_{1,\theta}', \mathtt{SK}_{2,\theta}, \mathtt{SK}_{2,\theta}', \mathtt{SK}_{3,\theta})\}_{\theta \in \mathsf{Path}(\mathtt{BT}, \eta)}$ を出力する．
- $\mathsf{KeyUp}(msk, st, RL, T)$: $msk = (\mathtt{MK}_1, \mathtt{MK}_2)$ とする．各ノード $\theta \in \mathsf{KUNode}(\mathtt{BT}, RL, T)$，$P_\theta$ が定義されていればそれを呼び出す．そうでなければ $P_\theta \xleftarrow{\$} \mathbb{G}_2$ を選び，θ に保存する．$s_\theta \xleftarrow{\$} \mathbb{Z}_p$ を選び，以下を計算する．

  $$
  \begin{aligned}
  & \mathtt{KU}_{1,\theta}' := P_\theta^{-1} \mathtt{MK}_1 \left((g_2^{y_4})^T g_2^{y_5}\right)^{s_\theta}, \\
  & \mathtt{KU}_{2,\theta}' := P_\theta^{-1} \mathtt{MK}_2 \left((g_2^{x_4})^T g_2^{x_5}\right)^{-s_\theta}, \quad \mathtt{KU}_{3,\theta} := g_2^{s_\theta}.
  \end{aligned}
  $$

  最終的に $ku_T := \{(\mathtt{KU}_{1,\theta}', \mathtt{KU}_{2,\theta}', \mathtt{KU}_{3,\theta})\}_{\theta \in \mathsf{KUNode}(\mathtt{BT}, RL, T)}$ を出力する．
- $\mathsf{DKGen}(sk_{\mathtt{I}}, ku_T)$: $sk_{\mathtt{I}} = \{(\mathtt{SK}_{1,\theta}, \mathtt{SK}_{1,\theta}', \mathtt{SK}_{2,\theta}, \mathtt{SK}_{2,\theta}', \mathtt{SK}_{3,\theta})\}_{\theta \in \Theta_{\mathrm{SK}}}$，$ku_T = \{(\mathtt{KU}_{1,\theta}', \mathtt{KU}_{2,\theta}', \mathtt{KU}_{3,\theta})\}_{\theta \in \Theta_{\mathrm{KU}}}$ とする．$\Theta_{\mathrm{SK}} \cap \Theta_{\mathrm{KU}} = \emptyset$ であれば $\perp$ を出力する．そうでなければ，ある $\theta \in \Theta_{\mathrm{SK}} \cap \Theta_{\mathrm{KU}}$ に対して $R, S \xleftarrow{\$} \mathbb{Z}_p$ を選

び，以下を計算する．

$$\mathtt{DK}_1 := \mathtt{SK}_{1,\theta}(g_2^{y_2})^R,\ \ \mathtt{DK}_2 := \mathtt{SK}_{2,\theta}(g_2^{x_2})^{-R},$$
$$\mathtt{DK}'_1 := \mathtt{SK}'_{1,\theta}\mathtt{KU}'_{1,\theta}\left((g_2^{y_1})^{\mathtt{I}}g_2^{y_3}\right)^R\left((g_2^{y_4})^T g_2^{y_5}\right)^S,$$
$$\mathtt{DK}'_2 := \mathtt{SK}'_{2,\theta}\mathtt{KU}'_{2,\theta}\left((g_2^{x_1})^{\mathtt{I}}g_2^{x_3}\right)^{-R}\left((g_2^{x_4})^T g_2^{x_5}\right)^{-S},$$
$$\mathtt{DK}_3 := \mathtt{SK}_{3,\theta}g_2^R,\ \ \mathtt{DK}_4 := \mathtt{KU}_{3,\theta}g_2^S.$$

$dk_{\mathtt{I},T} := (\mathtt{DK}_1, \mathtt{DK}'_1, \mathtt{DK}_2, \mathtt{DK}'_2, \mathtt{DK}_3, \mathtt{DK}_4)$ を出力する．

- $\mathsf{Enc}(M, \mathtt{I}, T)$: $t, \mathtt{tag} \xleftarrow{\$} \mathbb{Z}_p$ を選び，以下を計算する．

$$C_0 := Mz^t,\ \ C_1 := g_1^t,\ \ C_2 := (g_1^\alpha)^t,$$
$$C_3 := \left(u_1^{\mathtt{I}}w_1^{\mathtt{tag}}h_1\right)^t,\ \ C_4 := (v_1^T\hat{v}_1)^t.$$

$C_{\mathtt{I},T} := (C_0, C_1, C_2, C_3, C_4, \mathtt{tag})$ を出力する．

- $\mathsf{Dec}(dk_{\mathtt{I},T}, C_{\mathtt{I},T})$: $dk_{\mathtt{I},T} = (\mathtt{DK}_1, \mathtt{DK}'_1, \mathtt{DK}_2, \mathtt{DK}'_2, \mathtt{DK}_3, \mathtt{DK}_4)$ and $C_{\mathtt{I},T} = (C_0, C_1, C_2, C_3, C_4, \mathtt{tag})$ に対して，

$$M = \frac{C_0 e(C_3, \mathtt{DK}_3)e(C_4, \mathtt{DK}_4)}{e(C_1, \mathtt{DK}_1^{\mathtt{tag}}\mathtt{DK}'_1)e(C_2, \mathtt{DK}_2^{\mathtt{tag}}\mathtt{DK}'_2)},$$

を出力する．

- $\mathsf{Revoke}(\mathtt{I}, T, RL, st)$: $RL := RL \cup \{(\mathtt{I}, T)\}$ を出力する．

紙面の都合上，正当性を満たすことの確認は省略する．以下の定理を得る．証明は次節で行う．

定理 2. *ADDH1* 仮定及び *DDH2* 仮定が成り立つならば，上記構成法 Π は *IND-RID-CPA* 安全である．

4.1 提案構成法の安全性証明

以下の補題を示すことで，その系として定理 2 を示す．

補題 4. *3* 節の拡張 *JR-IBE* Π_{JR} の *IND-ID-CPA* 安全性の下で，提案した *RIBE* Π は *IND-RID-CPA* 安全である．

証明． 提案構成法 Π の IND-RID-CPA 安全性を破る PPT アルゴリズム $\mathcal{A}$ を用いて，拡張 JR-IBE Π_{JR} の IND-ID-CPA 安全性を破る PPT アルゴリズム $\mathcal{B}$ を構成する．

最初に $\mathcal{B}$ は Π_{JR} の公開パラメータ PP を受け取り，チャレンジの際に $\mathcal{A}$ が送ってくる T^* をランダムに推測する．$1/|\mathcal{T}|$ の確率で推測が成功する．もし $\mathcal{B}$ が推測が間違っていることに気付いたら，即座にシミュレーションを中止しランダムビット b' を送る．これ以降は，$\mathcal{B}$ の推測が当たっているものとして証明を進める．次に $\mathcal{B}$ は N 個の葉ノードを持つ BT を作る．$\tilde{x}, \hat{x}, \tilde{y}, \hat{y} \xleftarrow{\$} \mathbb{Z}_p$ を選び，

$$x_4 = \beta x_0 + \tilde{x},\ \ x_5 = -T^*\beta x_0 + \hat{x},\ \ y_4 = \beta y_0 + \tilde{y},$$
$$y_5 = -T^*\beta y_0 + \hat{y},\ \ -x_4\alpha + y_4 := -(\beta x_0 + \tilde{x})\alpha + \beta y_0 + \tilde{y},$$
$$-x_5\alpha + y_5 := -(-T^*\beta x_0 + \hat{x})\alpha - T^*\beta y_0 + \hat{y}$$

とし，以下のように mpk を計算し，$\mathcal{A}$ に送る．

$$g_2^{x_4} := g_2^{\beta x_0}g_2^{\tilde{x}},\ \ g_2^{x_5} := (g_2^{\beta x_0})^{-T^*}g_2^{\hat{x}},\ \ g_2^{y_4} := g_2^{\beta y_0}g_2^{\tilde{y}},$$
$$g_2^{y_5} := (g_2^{\beta y_0})^{-T^*}g_2^{\hat{y}},\ \ v_1 := \chi_1(g_1^\alpha)^{\tilde{x}}g_1^{\tilde{y}},\ \ \hat{v}_1 := \chi_1^{-T^*}(g_1^\alpha)^{\hat{x}}g_1^{\hat{y}}.$$

更に $\mathcal{B}$ は $\mathcal{A}$ が *SKGen* オラクルに $\mathtt{I}^*$ をクエリしてくるのか，またいつ *DKGen* オラクル（及び *SKGen* オラクル）にクエリしてくるのかをあらかじめ推測する．具体的には，q_1 をチャレンジまでの *SKGen* または *DKGen* オラクルへのクエリ回数とし，$\mathcal{B}$ はランダムに $(k^*, i^*) \in \{1,2\} \times \{1, 2, \ldots, q_1, q_1+1\}$ を推測する．$k^* = 1$ は $\mathcal{A}$ が $\mathtt{I}^*$ を *SKGen* オラクルにクエリすることを意味し，$k^* = 2$ はしないことを意味している．$i^* \in \{1, 2, \ldots, q\}$ は $\mathcal{A}$ が $\mathtt{I}^*$ を i^* 番目のクエリで $\mathcal{B}$ に *SKGen* または *DKGen* オラクルに問い合わせたことを示しており，$i^* = q_1 + 1$ は $\mathcal{A}$ が $\mathtt{I}^*$ に問い合わせるのがチャレンジ以降またはまったく問い合わせないことを意味している．$k^* = 1$ の時は $\mathtt{I}^*$ は必ず T^* の前に削除リストに登録されることに留意する．ここで [17] を基に攻撃者を分類する．各 $k^* \in \{1,2\}$ に対して，$i^* \in \{1, 2, \ldots, q\}$ のときタイプ k^*-a 攻撃者と呼び，$i^* = q_1 + 1$ のときタイプ k^*-b 攻撃者と呼ぶ．$1/2(q_1+1)$ の確率でこの推測は成功する．もし $\mathcal{B}$ の推測が間違っていることに気付いたら，即座にシミュレーションを中止しランダムビット b' を送る．これ以降は，$\mathcal{B}$ の推測が成功しているものとして証明を進める．

タイプ 1 攻撃者． タイプ 1-a 攻撃者とタイプ 1-b 攻撃者に対するシミュレーションの違いは，*SKGen* オラクル及び *DKGen* オラクルのシミュレート方法の違いのみである．$\mathcal{B}$ はまず，BT から $\mathtt{I}^*$ 用のノード η^* をあらかじめランダムに選んでおく．

タイプ 1-a 攻撃者の場合． $\mathcal{B}$ が j 番目の ID $\mathtt{I}$ を秘密鍵クエリ $\mathtt{I}$ または復号鍵クエリ $(\mathtt{I}, T)$ として受け取ったとする．それぞれ，以下のように $sk_{\mathtt{I}}$，または $dk_{\mathtt{I},T}$ を計算する．

Case $j < i^*$: $\mathcal{B}$ はまず *KeyGen* オラクルに $\mathtt{I}$ をクエリし，$SK_{\mathtt{I}} := (D_1, D'_1, D_2, D'_2, D_3)$ を得る．もし $\mathtt{I}$ が保存されていなければ，$\mathcal{B}$ は BT からランダムに空いている葉ノード η $(\neq \eta^*)$ を選び，$\mathtt{I}$ を保存する．

***SKGen* オラクル:** 各 $\theta \in \mathsf{Path}(\mathtt{BT}, \eta)$ に対して，P_θ が定義されていればそれを用い，そうでなければ $P_\theta \xleftarrow{\$} \mathbb{G}_2$ を選び，P_θ を θ に保存する．$\mathcal{B}$ は $r_\theta \xleftarrow{\$} \mathbb{Z}_p$ を選び，$\theta \notin \mathsf{Path}(\mathtt{BT}, \eta^*)$ ならば以下 $(*1)$ を計算する．

$$\mathtt{SK}_{1,\theta} := D_1(g_2^{y_2})^{r_\theta},\ \mathtt{SK}'_{1,\theta} := P_\theta D'_1\left((g_2^{y_1})^{\mathtt{I}}g_2^{y_3}\right)^{r_\theta},$$
$$\mathtt{SK}_{2,\theta} := D_2(g_2^{x_2})^{-r_\theta},\ \mathtt{SK}'_{2,\theta} := P_\theta D'_2\left((g_2^{x_1})^{\mathtt{I}}g_2^{x_3}\right)^{-r_\theta},$$
$$\mathtt{SK}_{3,\theta} := D_3 g_2^{r_\theta}.$$

そうでなければ，$\mathcal{B}$ は以下 $(*2)$ を計算する．

$$\mathtt{SK}_{1,\theta} := (g_2^{y_2})^{r_\theta},\ \mathtt{SK}'_{1,\theta} := P_\theta\left((g_2^{y_1})^{\mathtt{I}}g_2^{y_3}\right)^{r_\theta},$$
$$\mathtt{SK}_{2,\theta} := (g_2^{x_2})^{-r_\theta},\ \mathtt{SK}'_{2,\theta} := P_\theta\left((g_2^{x_1})^{\mathtt{I}}g_2^{x_3}\right)^{-r_\theta},$$

$\mathsf{SK}_{3,\theta} := g_2^{r_\theta}$.

$sk_{\mathtt{I}} := \{(\mathsf{SK}_{1,\theta}, \mathsf{SK}'_{1,\theta}, \mathsf{SK}_{2,\theta}, \mathsf{SK}'_{2,\theta}, \mathsf{SK}_{3,\theta})\}_{\theta \in \mathsf{Path}(\mathtt{BT},\eta)}$ を出力する．

***DKGen* オラクル:** $\mathcal{B}$ は $sk_{\mathtt{I}}$ を上記の要領で生成し（または保存してある $sk_{\mathtt{I}}$ を用い），DKGen アルゴリズムを実行，出力する．ここで，$\mathcal{A}$ は *DKGen* オラクルに $(\mathtt{I}, T)$ をクエリする前に *KeyUp* オラクルに T をクエリしていなければならないため，ku_T はこの時点で必ず生成されていることに留意する．

Case $j = i^*$**:** $\mathcal{B}$ は $\mathtt{I}^* := \mathtt{I}$ とし，$\mathtt{I}^*$ を η^* に保存する．

***SKGen* オラクル:** Case $j < i^*$ のように P_θ を選び，$(*2)$ のように計算した $sk_{\mathtt{I}^*}$ を出力する．

***DKGen* オラクル:** 上記のように $sk_{\mathtt{I}^*}$ を生成し $dk_{\mathtt{I}^*,T}$ を計算，出力する．

Case $j > i^*$**:** $\mathtt{I} \neq \mathtt{I}^*$ ならば Case $j < i^*$ と同様に，そうでなければ Case $j = i^*$ と同様にシミュレートする．

タイプ 1-b 攻撃者の場合． タイプ 1-b 攻撃者 $\mathcal{A}$ はチャレンジ後にのみ $\mathtt{I}^*$ に関するクエリを $\mathcal{B}$ に送るため，$\mathcal{B}$ はクエリを受け取った時点でその ID がターゲット ID かどうか識別が可能である．従って，$\mathtt{I} \neq \mathtt{I}^*$ の場合は Case $j < i^*$ と同様にシミュレートし，$\mathtt{I} = \mathtt{I}^*$ の場合は Case $j = i^*$ と同様にシミュレートすればよい．

以下はタイプ 1-a 攻撃者，タイプ 1-b 攻撃者同様に行う．

***KeyUp* オラクル.** 各 $\theta \in \mathsf{KUNode}(\mathtt{BT}, RL, T)$ に対して，P_θ が定義されていればそれを用い，そうでなければ $P_\theta \xleftarrow{\$} \mathbb{G}_2$ を選び，P_θ を θ に保存する．

$\mathcal{B}$ は $s_\theta \xleftarrow{\$} \mathbb{Z}_p$ を選び，$\theta \notin \mathsf{Path}(\mathtt{BT}, \eta^*)$ ならば以下 $(*3)$ を計算する．

$$\mathsf{KU}'_{1,\theta} := P_\theta^{-1}\left((g_2^{y_4})^T g_2^{y_5}\right)^{s_\theta},\quad \mathsf{KU}'_{2,\theta} := P_\theta^{-1}\left((g_2^{x_4})^T g_2^{x_5}\right)^{-s_\theta},$$

$$\mathsf{KU}_{3,\theta} := g_2^{s_\theta}.$$

そうでなければ以下を計算する．

$$\begin{aligned}
\mathsf{KU}'_{1,\theta} &:= P_\theta^{-1}((g_2^{y_4})^T g_2^{y_5})^{s_\theta}(g_2^{\frac{1}{\beta}})^{-\frac{T\tilde{y}+\hat{y}}{T-T^*}},\\
\mathsf{KU}'_{2,\theta} &:= P_\theta^{-1}((g_2^{x_4})^T g_2^{x_5})^{-s_\theta}(g_2^{\frac{1}{\beta}})^{\frac{T\tilde{x}+\hat{x}}{T-T^*}},\\
\mathsf{KU}_{3,\theta} &:= g_2^{s_\theta}(g_2^{\frac{1}{\beta}})^{-\frac{1}{T-T^*}}.
\end{aligned}$$

ここで，$sk_{\mathtt{I}^*}$ は必ず T^* までに失効されているため，$\theta \in \mathsf{Path}(\mathtt{BT}, \eta^*)$ かつ $\theta \in \mathsf{KUNode}(\mathtt{BT}, RL, T^*)$ となるような θ は存在しないことに留意する．最終的に $ku_T := \{(\mathsf{KU}'_{1,\theta}, \mathsf{KU}'_{2,\theta}, \mathsf{KU}_{3,\theta})\}_{\theta \in \mathsf{KUNode}(\mathtt{BT},RL,T)}$ を出力する．

上記の計算は正しいシミュレーションになっていることは，$s'_\theta := s_\theta - \frac{1}{(T-T^*)\beta}$ として以下のように確認できる．

$$\begin{aligned}
&((g_2^{y_4})^T g_2^{y_5})^{s_\theta}(g_2^{\frac{1}{\beta}})^{-\frac{T\tilde{y}+\hat{y}}{T-T^*}}\\
&= g_2^{y_0}(g_2^{(T-T^*)\beta y_0 + T\tilde{y}+\hat{y}})^{s_\theta}(g_2^{(T-T^*)\beta y_0 + T\tilde{y}+\hat{y}})^{-\frac{1}{(T-T^*)\beta}}\\
&= g_2^{y_0}(g_2^{(T-T^*)\beta y_0 + T\tilde{y}+\hat{y}})^{s'_\theta},\\
&((g_2^{x_4})^T g_2^{x_5})^{-s_\theta}(g_2^{\frac{1}{\beta}})^{\frac{T\tilde{x}+\hat{x}}{T-T^*}}\\
&= g_2^{-x_0}(g_2^{(T-T^*)\beta x_0 + T\tilde{x}+\hat{x}})^{-s_\theta}(g_2^{(T-T^*)\beta x_0 + T\tilde{x}+\hat{x}})^{\frac{1}{(T-T^*)\beta}}\\
&= g_2^{-x_0}(g_2^{(T-T^*)\beta x_0 + T\tilde{x}+\hat{x}})^{-s'_\theta},\\
&g_2^{s_\theta}(g_2^{\frac{1}{\beta}})^{-\frac{1}{T-T^*}} = g_2^{s_\theta - \frac{1}{(T-T^*)\beta}} = g_2^{s'_\theta}.
\end{aligned}$$

チャレンジ． $\mathcal{B}$ は $\mathcal{A}$ から $(M_0^*, M_1^*, \mathtt{I}^*, T^*)$ を受け取ったら，Π の IND-ID-CPA ゲームのチャレンジャーに $(M_0^*, M_1^*, \mathtt{I}^*)$ を送り，返ってきた $(C_0^*, C_1^*, C_2^*, C_3^*, \mathtt{tag}^*)$ を用い，$C_4^* := (C_2^*)^{-(T^*\tilde{x}+\hat{x})}(C_1^*)^{T^*\tilde{y}+\hat{y}}$ を計算する．正しい C_4^* であることは $C_4^* = (v_1^{T^*}\hat{v}_1)^t = g_1^{t(-T^*\tilde{x}\alpha + T^*\tilde{y} - \hat{x}\alpha + \hat{y})} = g_1^{-t\alpha(T^*\tilde{x}+\hat{x}) + t(T^*\tilde{y}+\hat{y})}$ から確認できる．$\mathcal{B}$ は $\mathcal{A}$ に $(C_0^*, C_1^*, C_2^*, C_3^*, C_4^*, \mathtt{tag}^*)$ を送る．

$\mathcal{B}$ は $\mathcal{A}$ の出力する b' をそのまま用いる．また，$\mathcal{B}$ の全ての振る舞いが $\mathcal{A}$ の観点から完璧なシミュレーションとなっていることは，[17] の Claim 1 と同様に示すことができる．

タイプ 2 攻撃者． タイプ 2-a 攻撃者とタイプ 2-b 攻撃者に対するシミュレーションの違いは，*DKGen* オラクルのシミュレート方法の違いのみとなる．*SKGen* オラクルについては $(*1)$ と同様の計算を行えばよく，*KeyUp* オラクルは $(*3)$ と同様に計算を行えばよい．

タイプ 2-a 攻撃者の場合． q_d $(\leq q_1)$ をチャレンジ前における *DKGen* オラクルへのクエリ数とする．$\mathcal{B}$ が j 番目の ID $\mathtt{I}$ を含む復号鍵クエリ $(\mathtt{I}, T)$ を受け取ったとき，以下のように $dk_{\mathtt{I},T}$ を生成する．

Case $j < i^*$**:** 上記のように $sk_{\mathtt{I}}$ と ku_T を生成し，DKGen アルゴリズムを実行する．

Case $j = i^*$**:** $r, s \xleftarrow{\$} \mathbb{Z}_p$ を選び，以下を計算，出力する．

$$\mathsf{DK}_1 := (g_2^{y_2})^r, \mathsf{DK}'_{1,\theta} := \left((g_2^{y_1})^{\mathtt{I}^*} g_2^{y_3}\right)^r ((g_2^{y_4})^T g_2^{y_5})^s (g_2^{\frac{1}{\beta}})^{-\frac{T\tilde{y}+\hat{y}}{T-T^*}},$$

$$\mathsf{DK}_2 := (g_2^{x_2})^{-r}, \mathsf{DK}'_2 := \left((g_2^{x_1})^{\mathtt{I}^*} g_2^{x_3}\right)^{-r} ((g_2^{x_4})^T g_2^{x_5})^{-s} (g_2^{\frac{1}{\beta}})^{\frac{T\tilde{x}+\hat{x}}{T-T^*}},$$

$$\mathsf{DK}_3 := g_2^r,\quad \mathsf{DK}_4 := g_2^s (g_2^{\frac{1}{\beta}})^{-\frac{1}{T-T^*}}.$$

Case $j > i^*$**:** $\mathtt{I} \neq \mathtt{I}^*$ であれば Case $j < i^*$ と同様に，そうでなければ Case $j = i^*$ と同様にシミュレートを行う．

タイプ 2-b 攻撃者の場合． タイプ 1-b 攻撃者同様，$\mathtt{I} \neq \mathtt{I}^*$ の場合はタイプ 2-a 攻撃者の Case $j < i^*$ と同様に，$\mathtt{I} = ID^*$ の場合は Case $j = i^*$ と同様にシミュレートする．

チャレンジ． タイプ 1 攻撃者の場合と同じようにシミュレートする．

$\mathcal{B}$ は $\mathcal{A}$ の出力する b' をそのまま用いる．また，$\mathcal{B}$ の全ての振る舞いが $\mathcal{A}$ の観点から完璧なシミュレーションとなっていることは，[17] の Claim 2 と同様に示すことができる．

表 1 提案構成法の効率.

方式	$\#mpk$	$\#msk$	$\#C$	$\#sk$	$\#ku$	$\#dk$	計算量仮定
SE13 [17]	$(6+n)\lvert\mathbb{G}_p\rvert$	$\lvert\mathbb{G}_p\rvert$	$3\lvert\mathbb{G}_p\rvert + \lvert\mathbb{G}_T^{\mathsf{sym}}\rvert$	$(2\log N)\lvert\mathbb{G}_p\rvert$	$(2r\log\frac{N}{r})\lvert\mathbb{G}_p\rvert$	$3\lvert\mathbb{G}_p\rvert$	DBDH
提案方式	$7\lvert\mathbb{G}_1\rvert + 11\lvert\mathbb{G}_2\rvert + \lvert\mathbb{G}_T^{\mathsf{asym}}\rvert$	$2\lvert\mathbb{G}_2\rvert$	$4\lvert\mathbb{G}_1\rvert + \lvert\mathbb{G}_T^{\mathsf{asym}}\rvert + \lvert\mathbb{Z}_p\rvert$	$(5\log N)\lvert\mathbb{G}_2\rvert$	$(3r\log\frac{N}{r})\lvert\mathbb{G}_2\rvert$	$6\lvert\mathbb{G}_1\rvert$	ADDH1, DDH2

$\lvert\mathbb{G}_1\rvert$, $\lvert\mathbb{G}_2\rvert$, $\lvert\mathbb{G}_T^{\mathsf{asym}}\rvert$ はそれぞれ $\mathbb{G}_1$, $\mathbb{G}_2$, $\mathbb{G}_T$ の要素のビット長を表し，$\lvert\mathbb{G}_p\rvert$, $\lvert\mathbb{G}_T^{\mathsf{sym}}\rvert$ はそれぞれ [17] で用いられている素数位数群 $\mathbb{G}_p$, $\mathbb{G}_T$ の要素のビット長を表す．$\lvert\mathbb{Z}_p\rvert$ は $\mathbb{Z}_p$ の要素のビット長を表す．$\#mpk$, $\#msk$, $\#C$, $\#sk$, $\#ku$, $\#dk$ はそれぞれ各パラメータの大きさを表す．N は最大ユーザ数，r は削除ユーザ数，n は ID のビット長を表し，例えば 32 バイトのメールアドレスを ID とするならば $n = 256$ となる．

最後に帰着ロスを見積もる．$\mathcal{E}_1$ を $\mathcal{B}$ が正しく T^* を推測するイベントとし，$\mathcal{E}_2$ を $\mathcal{B}$ が (k^*, i^*) を正しく推測するイベントとする．この時，

$$\begin{aligned} Adv_{\Pi,\mathcal{A}}^{IND\text{-}ID\text{-}CPA}(\lambda) &= \left|\Pr[b'=b] - \frac{1}{2}\right| \\ &= \frac{1}{|\mathcal{T}|}\left|\Pr[b'=b \mid \mathcal{E}_1] - \frac{1}{2}\right| \\ &= \frac{1}{2|\mathcal{T}|(q_1+2)}\left|\Pr[b'=b \mid \mathcal{E}_1 \wedge \mathcal{E}_2] - \frac{1}{2}\right| \\ &= \frac{1}{2|\mathcal{T}|(q_1+2)} Adv_{\Pi,\mathcal{A}}^{IND\text{-}RID\text{-}CPA}(\lambda, N). \end{aligned}$$

が成り立つ．従って，$Adv_{\Pi,\mathcal{A}}^{IND\text{-}RID\text{-}CPA}(\lambda, N) \leq 8|\mathcal{T}|(q_1+2)Adv_{\mathcal{G},\mathcal{B}}^{ADDH1}(\lambda) + 2|\mathcal{T}|q(q_1+2)Adv_{\mathcal{G},\mathcal{B}}^{DDH2}(\lambda)$．ただし，$q$ は $KeyGen$ オラクルへのクエリ数である． □

4.2 効率性比較

提案構成法と，適応的安全かつ復号鍵漏洩耐性を持ち，素数位数群上で構成された RIBE 方式 [17] との比較を表 1 に示す．提案方式は mpk の長さが ID の長さに依存しないため，非常に効率的である．また [17] の帰着効率が $O(n|\mathcal{T}|q^2)$ であるのに対して提案方式の帰着効率は $O(|\mathcal{T}|q_1 q)$ であり，更に提案方式は非対称ペアリングを用いているため，各群要素長（特に $\mathbb{G}_1$ の群要素長）に関しても効率的である．

謝辞 本研究は JSPS 科研費 16J10532, 16K00198 の助成によるものです．また本研究の初期段階において有益なコメントを頂きました，徐在弘氏に深く感謝いたします．

参考文献

[1] Boldyreva, A., Goyal, V. and Kumar, V.: Identity-based encryption with efficient revocation, *CCS*, ACM, pp. 417–426 (2008).
[2] Boneh, D. and Franklin, M.: Identity-Based Encryption from the Weil Pairing, *CRYPTO*, pp. 213–229 (2001).
[3] Chen, J., Lim, H. W., Ling, S., Su, L. and Wang, H.: Anonymous and Adaptively Secure Revocable IBE with Constant Size Public Parameters, CoRR, abs/1210.6441 (2012).
[4] Chen, J., Lim, H. W., Ling, S., Wang, H. and Nguyen, K.: Revocable Identity-Based Encryption from Lattices, *ACISP*, pp. 390–403 (2012).
[5] Cheng, S. and Zhang, J.: Adaptive-ID Secure Revocable Identity-Based Encryption from Lattices via Subset Difference Method, *ISPEC*, pp. 283–297 (2015).
[6] Emura, K., Seo, J. H. and Youn, T.: Semi-Generic Transformation of Revocable Hierarchical Identity-Based Encryption and Its DBDH Instantiation, *IEICE Transactions*, Vol. 99-A, No. 1, pp. 83–91 (2016).
[7] Ishida, Y., Watanabe, Y. and Shikata, J.: Constructions of CCA-Secure Revocable Identity-Based Encryption, *ACISP*, pp. 174–191 (2015).
[8] Jutla, C. and Roy, A.: Shorter Quasi-Adaptive NIZK Proofs for Linear Subspaces, *ASIACRYPT*, Springer, pp. 1–20 (2013).
[9] Lee, K. and Park, S.: Revocable Hierarchical Identity-Based Encryption with Shorter Private Keys and Update Keys, *IACR Cryptology ePrint Archive*, Vol. 2016, p. 460 (2016).
[10] Libert, B. and Vergnaud, D.: Adaptive-ID Secure Revocable Identity-Based Encryption, *CT-RSA*, Springer, pp. 1–15 (2009).
[11] Lin, H., Cao, Z., Fang, Y., Zhou, M. and Zhu, H.: How to design space efficient revocable IBE from non-monotonic ABE, *ASIACCS*, pp. 381–385 (2011).
[12] Naor, D., Naor, M. and Lotspiech, J.: Revocation and Tracing Schemes for Stateless Receivers, *CRYPTO*, pp. 41–62 (2001).
[13] Ramanna, S., Chatterjee, S. and Sarkar, P.: Variants of Waters' Dual System Primitives Using Asymmetric Pairings, *PKC*, Springer, pp. 298–315 (2012).
[14] Ramanna, S. and Sarkar, P.: Efficient (Anonymous) Compact HIBE from Standard Assumptions, *Provable Security*, Springer, pp. 243–258 (2014).
[15] Ryu, G., Lee, K., Park, S. and Lee, D. H.: Unbounded Hierarchical Identity-Based Encryption with Efficient Revocation, *WISA*, pp. 122–133 (2015).
[16] Seo, J. H. and Emura, K.: Efficient Delegation of Key Generation and Revocation Functionalities in Identity-Based Encryption, *CT-RSA*, Springer, pp. 343–358 (2013).
[17] Seo, J. H. and Emura, K.: Revocable Identity-Based Encryption Revisited: Security Model and Construction, *PKC*, Springer, pp. 216–234 (2013).
[18] Seo, J. H. and Emura, K.: Adaptive-ID Secure Revocable Hierarchical Identity-Based Encryption, *IWSEC*, Springer, pp. 21–38 (2015).
[19] Seo, J. H. and Emura, K.: Revocable Hierarchical Identity-Based Encryption: History-Free Update, Security Against Insiders, and Short Ciphertexts, *CT-RSA*, Springer, pp. 106–123 (2015).
[20] Su, L., Lim, H. W., Ling, S. and Wang, H.: Revocable IBE Systems with Almost Constant-Size Key Update, *Pairing-Based Cryptography*, pp. 168–185 (2013).
[21] Waters, B.: Efficient Identity-Based Encryption Without Random Oracles, *EUROCRYPT*, pp. 114–127 (2005).