Computer Security Symposium 2018
22 - 25 October 2018

# CBDH仮定に基づく効率的な閾値公開鍵暗号

海老名 将宏[1]　渡邉 洋平[2,†1]　四方 順司[1]

**概要**：暗号プロトコルをできる限り弱い計算量仮定に基づいて構成することは重要な研究指針の一つである．本稿では，計算問題である Computational Bilinear Diffie-Hellman (CBDH) 問題の困難性に基づく，選択暗号文攻撃 (CCA) に対して安全な閾値公開鍵暗号 (TPKE) の効率的な構成法を提案する．具体的には，CBDH 仮定に基づく閾値 ID ベース鍵カプセル化メカニズム (TIB-KEM) を提案し，CSS 2013 にて三田らによって提案された一般的構成法を用いることで CCA 安全な TPKE 方式を得る．提案構成法は計算問題の困難性の下での CCA 安全性と定数倍長のパラメータを達成する初めての TPKE である．

**キーワード**：ハイブリッド暗号, 閾値暗号 , CBDH 仮定

# Efficient Threshold Public-Key Encryption from the CBDH Assumption

Masahiro Ebina[1]　Yohei Watanabe[2,†1]　Junji Shikata[1]

***Abstract:*** It is one of important research directions to design cryptographic protocols from the weakest possible complexity assumption. In this paper, we propose an efficient threshold public key encryption (TPKE) system secure against chosen ciphertext attacks (CCA) from the computational bilinear Diffie-Hellman (CBDH) assumption. Specifically, we propose a threshold identity-based key encapsulation mechanism (TIB-KEM) based on the CBDH assumption, and we obtain a CCA-secure TPKE scheme by applying a generic construction proposed by Mita et al. at CSS 2013. Our construction is the first TPKE scheme that simultaneously achieves constant-size parameters and CCA security from computational assumptions.

***Keywords:*** Hybrid encryption, threshold encryption, the CBDH assumption

## 1. はじめに

選択暗号文攻撃 (CCA) に対して安全な公開鍵暗号 (Public-Key Encryption: PKE) 方式について，これまでに多くの研究が行われてきた．特に，[9] で CCA 安全かつランダムオラクルを用いない PKE 方式が提案され，[10] でその構成が一般化されて以降，スタンダードモデルのおける CCA 安全な PKE に関して，様々な方式が提案されてきた．PKE を構成するには何らかの計算量仮定が必要となるが，計算量仮定が破れれば，その仮定に基づく暗号技術の安全性はただちに崩れてしまう．従って，できる限り弱い計算量仮定に基づいて暗号技術を構成するのは現代暗号理論分野における重要な研究方針のひとつである．さて，[9], [10] 等，これまで提案されてきた PKE 方式の多くが判定問題の困難性に基づくものである．一般的に判定問題から計算問題に帰着可能であることは知られているが，その逆は明らかになっておらず，計算問題の困難性に基づいた (効率的な) PKE を構成できるのか定かではなかったが，2004 年に [8] で CCA 安全かつランダムオラクルを用いない PKE の Computational Bilinear Diffie-Hellman

[1] 横浜国立大学大学院環境情報学府/研究院
Graduate School of Environment and Information Sciences, Yokohama National University
[2] 電気通信大学 大学院情報理工学研究科 日本学術振興会特別研究員 (PD)
JSPS Research Fellow (PD), Graduate School of Informatics and Engineering, The University of Electro-Communications
[†1] 現在，国立研究開発法人 情報通信研究機構 (NICT)
Presently with National Institute of Information and Communications Technology

(CBDH) 仮定に基づいた構成が提案された．その後，様々な PKE が計算問題の困難性に基づいて構成された．

本稿で扱う閾値公開鍵暗号 (Threshold Public-Key Encryption: TPKE) では，秘密鍵が $n$ 個に分割されており，各秘密鍵による暗号文の部分復号情報を閾値個以上集めて初めて平文を得ることができる PKE である．すなわち，閾値個未満の秘密鍵が漏洩したとしても，それを用いて暗号文を解読することができない．これまでに様々な TPKE が提案されてきたが (例えば [2], [4], [7])，(1) スタンダードモデルで証明可能で，(2) 計算問題の困難性の下で CCA 安全性を満たし，(3) 全パラメータが定数倍長であるような TPKE は未だに存在しない．本稿では，これら (1)–(3) を同時に達成する TPKE 方式を初めて提案する．

**関連研究．** TPKE に関する関連研究をいくつかまとめる．スタンダードモデルにおいて CCA 安全な TPKE 方式は [7] で初めて提案された．彼らの方式は Decisional Diffie-Hellman (DDH) 仮定の下で安全である．TPKE において，受信者と復号サーバが対話的に振る舞うことなく，公開鍵のみで暗号文の正当性検証が可能であることが望ましいが，[7] では秘密鍵を有する受信者と対話的に検証を行う必要があった．その後，[4] において，双線形写像を用いることで，非対話かつ公開鍵のみで検証可能な TPKE 方式が提案された．[3] で提案された ID ベース暗号 (Identity-Based Encryption: IBE) と，IBE から CCA 安全な PKE への変換として知られる CHK 変換 [8] を応用することで，Decisional Bilinear Diffie-Hellman (DBDH) 仮定の下で CCA 安全な TPKE を構成している．[11] では，CCA 安全な多重暗号 (Multiple Encryption: ME) の一般的構成が示されたと共に，ME から TPKE への変換が提案されている．従って，構成要素である PKE や IBE に計算問題の困難性に基づく方式を用いることで，計算問題の困難性に基づき，かつスタンダードモデルで CCA 安全な TPKE 方式を得ることができるが，公開鍵長や暗号文長がサーバ数 $n$ に依存しており，効率性に欠ける．

また，閾値鍵カプセル化メカニズム (Threshold Key Encapsulation Mechanism: TKEM) についても研究が行われている．[6] では，[4] と同様のアプローチを取りつつも，具体的に構成することで，[4] より効率的な構成を実現している．[2] では，CCA 安全な閾値タグ鍵カプセル化メカニズム (Threshold Tag-KEM: TTKEM) とワンタイム安全なデータカプセル化メカニズム (Data Encapsulation Mechanism: DEM) を組み合わせることで CCA 安全な TPKE が実現できることが示されている．これまで，CCA 安全な TPKE を構成するためには，CCA 安全な TKEM と CCA 安全な閾値データカプセル化メカニズム (Threshold DEM: TDEM) が必要であり，一般的に CCA 安全な TDEM は構成が難しいとされていた．[14] では，[2] を基にした CCA 安全な TTKEM の一般的構成が提案されている．Computational Diffie-Hellman (CDH) 仮定に基づいた構成法も示されているが，ランダムオラクルを用いる必要がある．[1] では，スタンダードモデルにおいて CCA 安全な TTKEM の一般的構成が示されている．具体的には，閾値 ID ベース鍵カプセル化メカニズム (Threshold Identity-based KEM: TIB-KEM) とワンタイム署名 (One-Time Signature: OTS) を用い，CHK 変換を応用する形で TTKEM を構成している．それまでの TIB-KEM の安全性は CCA 安全性しか知られていなかったため，彼らは CPA 安全性を新たに定義し，その安全性を満たす TIB-KEM を用いて安全性証明を行っている．

**本稿の成果．** [1] の構成法を基に，上記 (1)–(3) を全て満たす TPKE 方式を初めて構成する．具体的には，公開鍵，秘密鍵，暗号文の全てが定数倍長を達成し，かつスタンダードモデルで CBDH 仮定の下で CCA 安全である TPKE 方式を提案する．まず，3 節にて，TIB-KEM の CPA 安全性より弱いものとして，COA 安全性を定義する．次に，4 節にて，閾値型ではない通常の IB-KEM[16] の構成を基に，CBDH 仮定の下で COA 安全な TIB-KEM を提案し，またこの構成が CPA 安全性を満たさないことも示す．これはすなわち提案構成法が十分に効率化されていることを意味している．更に，5 節にて，[1] における TTKEM の一般的構成には CPA 安全性より弱い TIB-KEM を用いれば十分であることを指摘する．[1] では CHK 変換が応用されているが，一般的に CHK 変換で必要とされている CPA 安全性ではなく，より弱い COA 安全性で十分であることから，興味深い結果と言える．この原因として，TPKE における安全性定義が通常の PKE のそれと少し異なることが考えられる．最終的に，[2] で提案された一般的構成を適用することで，CBDH 仮定の下で CCA 安全かつランダムオラクルを用いない TPKE 方式を得ることができる．6 節にて既存方式との効率性の比較を行い，7 節にてまとめと今後の課題を述べる．

## 2. 準備

本節では準備として，TPKE, TTKEM，ワンタイム署名 (One-Time Signature: OTS)，計算量仮定について述べる．

### 2.1 Threshold Public-Key Encryption

**定義 1** (Threshold Public Key Encryption: TPKE)**.** *TPKE* は，以下の 4 つのアルゴリズム $\mathcal{TPKE} = (\mathsf{Setup}, \mathsf{Enc}, \mathsf{PartialDec}, \mathsf{Combine})$ から構成される．

$(\mathsf{pk}, \mathbf{sk}) \leftarrow \mathsf{Setup}(1^\kappa, k, n)$**:** セキュリティパラメータ $\kappa$，閾値 $k$，サーバ数 $n$ を入力として，公開鍵 $\mathsf{pk}$ と各サーバの秘密鍵 $\mathbf{sk} := (\mathsf{sk}_1, \ldots, \mathsf{sk}_n)$ を出力する．

$c \leftarrow \mathsf{Enc}(\mathsf{pk}, m)$**:** 公開鍵 $\mathsf{pk}$ と平文 $m \in \mathcal{M}_{\mathrm{TPKE}}$ を入力と

して，暗号文 $c$ を出力する．ただし，$\mathcal{M}_{\mathrm{TPKE}}$ は平文集合であり，公開鍵 $\mathsf{pk}$ に依存して定まる．

$\delta_i/\perp \leftarrow \mathsf{PartialDec}(\mathsf{pk}, \mathsf{sk}_i, i, c)$: 公開鍵 $\mathsf{pk}$，秘密鍵 $\mathsf{sk}_i$，サーバインデックス $i$ と暗号文 $c$ を入力として，部分復号情報 $\delta_i$ または復号失敗を表すシンボル $\perp$ を出力する．

$m/\perp \leftarrow \mathsf{Combine}(\mathsf{pk}, c, \{\delta_i\}_{i\in\mathcal{I}})$: 公開鍵 $\mathsf{pk}$，暗号文 $c$ と $\mathcal{I} \subset [n]$ に対応するサーバの部分復号情報 $\{\delta_i\}_{i\in\mathcal{I}}$ を入力として，その復号結果を出力する．ここで，$\perp$ は復号失敗を表す．

$\mathcal{TPKE}$ は次の正当性を満たす．全ての $\kappa \in \mathbb{N}$, 全ての $k \leq n$ である $k = \mathrm{poly}(\kappa), n = \mathrm{poly}(\kappa)$, 全ての $(\mathsf{pk}, \mathbf{sk}) \leftarrow \mathsf{Setup}(1^\kappa, k, n)$, 全ての $m \in \mathcal{M}_{\mathrm{TPKE}}$, 全ての $c \leftarrow \mathsf{Enc}(\mathsf{pk}, m)$, 全ての $|\mathcal{I}| \geq k$ である $\mathcal{I} \subset [n]$ に対して, $\mathsf{Combine}(\mathsf{pk}, c, \{\mathsf{PartialDec}(\mathsf{pk}, \mathsf{sk}_i, i, c)\}_{i\in\mathcal{I}}) = m$.

次に，TPKE の IND-CCA 安全性を定義する．本稿では，攻撃者が初めにコラプトするサーバを決定する static corruption を考える．攻撃者 $\mathcal{A}$ に対して，以下の IND-CCA ゲーム $\mathsf{Exp}^{\mathsf{CCA}}_{\mathcal{TPKE},\mathcal{A}}(\kappa, k, n)$ を考える．

Step 1. $\mathcal{A}$ はコラプトするサーバ集合 (のインデックス) $\mathcal{I}_c = \{\ell_1, ..., \ell_j\}$ を決定する．ここで $\mathcal{I}_c$ は $|\mathcal{I}_c| \leq k-1$ を満たす．

Step 2. チャレンジャーは $\mathsf{Setup}(1^\kappa, k, n)$ を実行し，$(\mathsf{pk}, \{\mathsf{sk}_i\}_{i\in\mathcal{I}_c})$ を $\mathcal{A}$ に渡す．残りの秘密鍵は保持する．

Step 3. $\mathcal{A}$ は PartialDec オラクルへ任意の多項式回数問い合わせを行う．PartialDec オラクルはクエリ $(c, i)$ に対して $\mathsf{PartialDec}(\mathsf{pk}, \mathsf{sk}_i, i, c)$ を返すオラクルである．

Step 4. $\mathcal{A}$ はチャレンジャーに $m_0, m_1$ を渡し，チャレンジャーは $\beta \xleftarrow{\$} \{0,1\}$ を一様ランダムに選び，$c^* \leftarrow \mathsf{Enc}(\mathsf{pk}, m_\beta)$ を $\mathcal{A}$ に返す．

Step 5. $\mathcal{A}$ は再び Step 3 を実行する．ただし PartialDec オラクルへのクエリ $(c, i)$ は $c \neq c^*$ であるものに制限される．

Step 6. $\mathcal{A}$ は $\beta' \in \{0,1\}$ をチャレンジャーへ送る．チャレンジャーは，もし $\beta' = \beta$ ならば 1 を出力し，そうでなければ 0 を出力する．

上記のゲームにおいて $\beta' = \beta$ であるとき攻撃者 $\mathcal{A}$ の勝利である．$\mathcal{A}$ のアドバンテージを $\mathsf{Adv}^{\mathsf{CCA}}_{\mathcal{TPKE},\mathcal{A}}(\kappa, k, n) = 2 \cdot \Pr[\mathsf{Exp}^{\mathsf{CCA}}_{\mathcal{TPKE},\mathcal{A}}(\kappa, k, n) = 1] - 1$ とする．

**定義 2** (IND-CCA). ある $\kappa_0 \in \mathbb{N}$ が存在して，全ての $\kappa \geq \kappa_0$, $k = \mathrm{poly}(\kappa)$, $n = \mathrm{poly}(\kappa)$ $(k \leq n)$ に対して，全ての確率的多項式時間アルゴリズム $\mathcal{A}$ のアドバンテージ $\mathsf{Adv}^{\mathsf{CCA}}_{\mathcal{TPKE},\mathcal{A}}(\kappa, k, n)$ が $\kappa$ に関して無視できるほど小さい時，$TPKE$ $\mathcal{TPKE}$ は IND-CCA 安全であるという．

### 2.2 Threshold tag-KEM

**定義 3** (Threshold Tag-KEM: TTKEM). *Threshold Tag-KEM* は，以下の 5 つのアルゴリズム $\mathcal{TTKEM} = (\mathsf{Setup}, \mathsf{SessionKeyGen}, \mathsf{Encap}, \mathsf{PartialDecap}, \mathsf{Reconst})$ から構成される．

$(\mathsf{ek}, \mathbf{dk}) \leftarrow \mathsf{Setup}(1^\kappa, k, n)$:セキュリティパラメータ $\kappa$，閾値 $k$，サーバ数 $n$ を入力として，公開鍵 $ek$ とサーバ毎の秘密鍵 $\mathbf{dk} := (\mathsf{dk}_1, ..., \mathsf{dk}_n)$ を出力する．

$(s, K) \leftarrow \mathsf{SessionKeyGen}(\mathsf{ek})$:公開鍵 $\mathsf{ek}$ を入力として，秘密情報 $s$ とセッション鍵 $K \in \mathcal{K}$ を出力する．ここで，$\mathcal{K}$ は $\mathsf{ek}$ によって決まる鍵空間である．

$h \leftarrow \mathsf{Encap}(s, \tau)$:秘密情報 $s$ とタグ $\tau \in \mathcal{T}$ を入力として，暗号文 $h$ を出力する．ここで $\mathcal{T}$ はタグ空間である．

$h_i/\perp \leftarrow \mathsf{PartialDecap}(\mathsf{ek}, \mathsf{dk}_i, h, \tau)$:公開鍵 $\mathsf{ek}$，秘密鍵 $\mathsf{dk}_i$，暗号文 $h$ とタグ $\tau$ を入力として，部分復号情報 $h_i$ または復号失敗を表すシンボル $\perp$ を出力する．

$\mathcal{K}/\perp \leftarrow \mathsf{Reconst}(\mathsf{ek}, h, (h_i)_{i\in\mathcal{I}}, \tau)$:公開鍵 $\mathsf{ek}$, 暗号文 $h$, $\mathcal{I}_1 \subset [N]$ に対応するサーバの部分復号情報 $\{h_i\}_{i\in\mathcal{I}}$ とタグ $\tau$ を入力として，その復号結果を出力する．ここで $\perp$ は復号失敗を表す．

$\mathcal{TTKEM}$ は次の正当性を満たす．全ての $\kappa \in \mathbb{N}$, 全ての $k \leq n$ である $k = \mathrm{poly}(\kappa), n = \mathrm{poly}(\kappa)$, 全ての $(\mathsf{ek}, \mathbf{dk}) \leftarrow \mathsf{Setup}(1^\kappa, k, n)$, 全ての $(s, K) \leftarrow \mathsf{SessionKeyGen}(\mathsf{ek})$, 全ての $h \leftarrow \mathsf{Encap}(s, \tau)$, 全ての $\tau \in \mathcal{T}$, 全ての $|\mathcal{I}| \geq k$ である $\mathcal{I} \subset [n]$ に対して, $\mathsf{Reconst}(\mathsf{ek}, h, \{\mathsf{PartialDecap}(\mathsf{ek}, \mathsf{dk}_i, h, \tau)_{i\in\mathcal{I}}\}, \tau) = K$ を満たすとする．

ここで TTKEM の IND-TCCA 安全性を定義する．攻撃者 $\mathcal{A}$ に対して，以下の IND-TCCA ゲームを考える．

Step 1. $\mathcal{A}$ はコラプトするサーバ集合 (のインデックス)$\mathcal{I}_c = \{\ell_1, ..., \ell_j\}$ を決定する．ここで $\mathcal{I}_c$ は $|I_c| \leq k-1$ を満たす．

Step 2. チャレンジャーは $\mathsf{Setup}(1^\kappa, k, n)$ アルゴリズムを実行し，$(\mathsf{ek}, \{\mathsf{dk}_i\}_{i\in\mathcal{I}_c})$ を $\mathcal{A}$ に渡す．残りの秘密鍵は保持する．

Step 3. $\mathcal{A}$ は PartialDec オラクルへ任意の多項式回数だけ問い合わせを行う．PartialDec オラクルはクエリ $(h, i)$ に対して $\mathsf{PartialDec}(\mathsf{ek}, \mathsf{dk}_i, h, \tau)$ を返す．

Step 4. チャレンジャーは $\mathsf{SessionKeyGen}(\mathsf{ek})$ を実行し，$(s^*, K_1^*)$ を得た後に，$K_0^* \xleftarrow{\$} \mathcal{K}$ を一様ランダムに選ぶ．そして $\beta \xleftarrow{\$} \{0,1\}$ を一様ランダムに選び，$K_\beta^*$ を $\mathcal{A}$ に渡す．

Step 5. $\mathcal{A}$ は Step 3 を実行する．

Step 6. $\mathcal{A}$ はチャレンジタグ $\tau^*$ をチャレンジャーに送る．

Step 7. チャレンジャーは $\mathsf{Encap}(s^*, \tau^*)$ アルゴリズムを実行し，$h^*$ を $\mathcal{A}$ に渡す．

Step 8. $\mathcal{A}$ は再び Step 3 を実行する．ただし，クエリ $(h, i)$ は $h \neq h^*$ に制限される．

Step 9. $\mathcal{A}$ は $\beta' \in \{0,1\}$ を出力する．チャレンジャーは，もし $\beta' = \beta$ ならば 1 を出力し，そうでなければ 0 を出力する．

上記のゲームにおいて $\beta' = \beta$ であるとき攻撃者 $\mathcal{A}$ の勝利である．$\mathcal{A}$ のアドバンテージを $Adv^{\mathsf{TCCA}}_{\mathcal{TTKEM},\mathcal{A}}(\kappa,k,n) = 2\cdot\Pr[\mathsf{Exp}^{\mathsf{TCCA}}_{\mathcal{TTKEM},\mathcal{A}}(\kappa,k,n)=1]-1$ とする．

**定義 4.** ある $\kappa_0 \in \mathbb{N}$ が存在して，全ての $\kappa \geq \kappa_0$, $k = \mathrm{poly}(\kappa)$, $n = \mathrm{poly}(\kappa)$ $(k \leq n)$ に対して，全ての確率的多項式時間アルゴリズム $\mathcal{A}$ のアドバンテージ $\mathsf{Adv}^{\mathsf{TCCA}}_{\mathcal{TTKEM},\mathcal{A}}(\kappa,k,n)$ が $\kappa$ に関して無視できるほど小さい時，*TTKEM* $\mathcal{TTKEM}$ は IND-TCCA 安全であるという．

### 2.3 ワンタイム署名 (OTS)

ワンタイム署名 (OTS:one-time signature) は以下の 3 つのアルゴリズム $\mathcal{OTS} = (\mathsf{KG}, \mathsf{Sig}, \mathsf{Vrfy})$ から成る．

$(\mathsf{sgk},\mathsf{vrk}) \leftarrow \mathsf{KG}(1^\kappa)$:セキュリティパラメータ $\kappa$ を入力として，署名鍵 sgk，検証鍵 vrk を出力する．

$\sigma \leftarrow \mathsf{Sig}(m,\mathsf{sgk})$:平文 $m \in \mathcal{M}$ と署名鍵 sgk を入力として，署名 $\sigma$ を出力する．

$0/1 \leftarrow \mathsf{Vrfy}(m,\sigma,\mathsf{vrk})$:平文 $m$，署名 $\sigma$，検証鍵 vrk を入力として，検証結果を出力する．

$\mathcal{OTS}$ は次の正当性を満たす．全ての $\kappa \in \mathbb{N}$，全ての $(\mathsf{sgk},\mathsf{vrk}) \leftarrow \mathsf{KG}(1^\kappa,m)$ に対して，$\mathsf{Vrfy}(m,\mathsf{Sig}(m,\mathsf{sgk}),\mathsf{vrk}) = 1$．sUF-CMA 安全性については紙面の都合上省略する．

### 2.4 CBDH 仮定

$p$ を $\kappa$ ビットの素数とする．生成元を $g$ とする位数が素数 $p$ の巡回群 $\mathbb{G}$ に対して，以下のような双線形写像 (ペアリング) $e : \mathbb{G}\times\mathbb{G} \to \mathbb{G}_T$ を考える．

- 双線形性：任意の $(u,v) \in \mathbb{G}\times\mathbb{G}$ および $a,b \in \mathbb{Z}_p$ に対して，$e(u^a,v^b) = e(u,v)^{ab}$ が成り立つ．
- 非退化性：$e(g,g) \neq 1 \in \mathbb{G}_T$ が成り立つ．
- 計算容易性：任意の $(u,v) \in \mathbb{G}\times\mathbb{G}$ に対して，$e(u,v)$ を効率的に計算できるアルゴリズムが存在する．

ランダムに選んだ $a,b,c \in \mathbb{Z}_p$ に対して，$A := g^a$, $B := g^b$, $C := g^c$ としたとき，$A,B,C$ から $\mathsf{bdh}(A,B,C) := e(g,g)^{abc}$ を求める問題を，CBDH 問題という．任意の確率的多項式時間アルゴリズム $\mathcal{A}$ のアドバンテージを $\mathsf{Adv}^{\mathsf{CBDH}}_{\mathcal{A}}(\kappa) := \Pr[\mathsf{bdh}(A,B,C) = T \mid A,B,C \xleftarrow{\$} \mathbb{G}, T \leftarrow \mathcal{A}(\kappa,g,A,B,C)]$ とする．

**定義 5** (CBDH 仮定)．ある $\kappa_0 \in \mathbb{N}$ が存在して，任意の $\kappa \geq \kappa_0$ に対して，任意の確率的多項式時間アルゴリズム $\mathcal{A}$ のアドバンテージ $\mathsf{Adv}^{\mathsf{CBDH}}_{\mathcal{A}}(\kappa)$ が $\kappa$ に関して無視できるほど小さいとき，CBDH 仮定が成立するという．

さらに [12] で示されている Goldreich-Levin 定理から，$f_{\mathrm{GL}} : \mathbb{G}_T \times \{0,1\}^u \to \{0,1\}$ のような一方向関数に関して，以下の定義と補題が与えられる．ランダムに選んだ $a,b,c \in \mathbb{Z}_p$ に対して，$A := g^a$, $B := g^b$, $C := g^c$ として，$R \xleftarrow{\$} \{0,1\}^u$ とする．ここで $u$ は，$\mathbb{G}_T$ の元のビット長を表す．さらに $K = f_{\mathrm{GL}}\Big(\mathsf{bdh}(A,B,C),R\Big)$ として，$U \xleftarrow{\$} \{0,1\}$ をランダムに選んだ時，$A,B,C$ から $\triangle_{\mathsf{bdh}} := (g,A,B,C,K,R)$ と $\triangle_{\mathsf{rand}} := (g,A,B,C,U,R)$ を識別する問題を GL-DBDH 問題という．任意の確率的多項式時間アルゴリズム $\mathcal{B}$ を $b \in \{0,1\}$ を出力する GL-DBDH 問題を解くアルゴリズムとして，$\mathcal{B}$ のアドバンテージを $\mathsf{Adv}^{\mathsf{GL\text{-}DBDH}}_{\mathcal{B}}(\kappa) := \Pr[\mathcal{B}(\kappa,g,A,B,C,\triangle_{\mathsf{bdh}}) = 0] - \Pr[\mathcal{B}(\kappa,g,A,B,C,\triangle_{\mathsf{rand}}) = 0]$ とする．

**定義 6.** ある $\kappa_0 \in \mathbb{N}$ が存在して，任意の $\kappa \geq \kappa_0$ に対して，任意の確率的多項式時間アルゴリズム $\mathcal{A}$ のアドバンテージ $\mathsf{Adv}^{\mathsf{GL\text{-}DBDH}}_{\mathcal{A}}(\kappa)$ が $\kappa$ に関して無視できるほど小さいとき，GL-DBDH 仮定が成立するという．

**補題 1.** ランダムに選んだ $a,b,c \in \mathbb{Z}_p$ に対して，$A := g^a$, $B := g^b$, $C := g^c$ として，$R \xleftarrow{\$} \{0,1\}^u$ とする．さらに $K = f_{gl}\Big(bdh(A,B,C),R\Big)$ として，$U \xleftarrow{\$} \{0,1\}$ をランダムに選んだ時，$A,B,C$ から $\triangle_{\mathsf{bdh}} = (g,A,B,C,K,R)$ と $\triangle_{\mathsf{rand}} = (g,A,B,C,U,R)$ を識別する多項式時間アルゴリズム $\mathcal{B}$ が存在するとき，CBDH 問題を破るような多項式時間アルゴリズムが存在する．

## 3. Threshold Identity-Based KEM

本節では，Threshold Identity-Based KEM (TIB-KEM) のモデルを述べたあと，新たな安全性定義として IND-sID-TCOA 安全性を与え，IND-sID-CPA 安全性との違いについても述べる．

### 3.1 TIB-KEM: モデル

**定義 7** (TIBKEM)．*TIB-KEM* は以下の 6 つのアルゴリズム $\mathcal{TIBKEM} = (\mathsf{Init},\mathsf{KeyShare},\mathsf{IBEncap},\mathsf{DecShare},\mathsf{DecVrfy},\mathsf{DecCombine})$ から構成される．

$(\mathsf{mpk},\mathbf{msk},\mathbf{vk}) \leftarrow \mathsf{Init}(1^\kappa,k,n)$: セキュリティパラメータ $\kappa$，閾値 $k$, サーバ数 $n$ を入力として，マスター公開鍵 mpk,, サーバ毎の秘密鍵と検証鍵 $\mathbf{msk} := (\mathsf{msk}_1,...,\mathsf{msk}_n), \mathbf{vk} := (\mathsf{vk}_1,...,\mathsf{vk}_n)$ を出力する．

$\mathsf{sk}[\mathtt{ID}]_i \leftarrow \mathsf{KeyShare}(\mathsf{mpk},\mathsf{msk}_i,i,\mathtt{ID})$: マスター公開鍵 mpk，サーバインデックス $i$，ユーザ *ID* ID と秘密鍵 $\mathsf{sk}_i$ を入力として，ユーザ *ID* に対応した秘密鍵 $\mathsf{sk}[\mathtt{ID}]_i$ を出力する．

$(c,K) \leftarrow \mathsf{IBEncap}(\mathsf{mpk},\mathtt{ID})$: マスター公開鍵 mpk とユーザ *ID* ID を入力として，セッション鍵 $K$ をランダムに生成し，ユーザ *ID* に対応した暗号文 $c$ と共に出力する．

$(i,c_i)/\perp \leftarrow \mathsf{DecShare}(\mathsf{mpk},i,\mathtt{ID},\mathsf{sk}[\mathtt{ID}]_i,c)$: マスター公開鍵 mpk，サーバインデックス $i$，ユーザ *ID* ID，秘密

鍵 $\mathsf{sk}[\mathtt{ID}]_i$ と暗号文 $c$ を入力として，サーバ $i$ の部分復号情報 $c_i$ を出力する．

$0/1 \leftarrow \mathsf{DecVrfy}(\mathsf{mpk}, \mathbf{vk}, i, \mathtt{ID}, c_i, c)$: マスター公開鍵 mpk，検証鍵 vk，サーバインデックス $i$，ユーザ $ID$ ID，部分復号情報 $c_i$ と暗号文 $c$ を入力として，$c_i$ の検証結果を出力する．

$K/\perp \leftarrow \mathsf{DecCombine}(\mathsf{mpk}, \mathbf{vk}, \mathtt{ID}, (c_i)_{i\in\mathcal{I}}, c)$: マスター公開鍵 mpk，検証鍵 vk，ユーザ $ID$ ID，$\mathcal{I} \subset [n]$ に対応するサーバの部分復号情報 $c_i$ と暗号文 $c$ を入力として，その復号結果を出力する．ここで $\perp$ は復号失敗を示す．

$\mathcal{TIBKEM}$ は次の正当性を満たす．全ての $\kappa \in \mathbb{N}$，全ての，$k \leq n$ である $k = \mathrm{poly}(\kappa), n = \mathrm{poly}(\kappa)$, 全ての $(\mathsf{mpk}, \mathbf{msk}, \mathbf{vk}) \leftarrow \mathsf{Init}(1^\kappa, k, n)$，全ての ID，全ての $i \in [n]$，全ての $\mathsf{sk}[\mathtt{ID}]_i \leftarrow \mathsf{KeyShare}(\mathsf{mpk}, \mathsf{msk}_i, i, \mathtt{ID})$，全ての $(c, K) \leftarrow \mathsf{IBEncap}(\mathsf{mpk}, \mathtt{ID})$，全ての $|\mathcal{I}| \geq k$ である $\mathcal{I} \subset [n]$ に対して，$\mathsf{DecCombine}(\mathsf{mpk}, \mathbf{vk}, \mathtt{ID}, \{\mathsf{DecShare}(\mathsf{mpk}, i, \mathtt{ID}, \mathsf{sk}[\mathtt{ID}]_i, c\}_{i\in\mathcal{I}}, c) = K$ を満たすとする．

### 3.2 安全性定義

ここでは，既存の TIB-KEM の安全性より弱い安全性である，Ciphertext Only Attack に対する安全性 (IND-sID-TCOA 安全性) を定義する．攻撃者 $\mathcal{A}$ に対して，以下のような IND-sID-TCOA ゲームを考える．

Step 1. $\mathcal{A}$ はコラプトするサーバ集合 (のインデックス) $\mathcal{I}_c = \{\ell_1, ..., \ell_j\}$ を決定する．ここで $\mathcal{I}_c$ は $|I_c| \leq k-1$ を満たす．

Step 2. チャレンジャーは $\mathsf{Init}(1^\kappa, k, n)$ を実行し，$(\mathsf{mpk}, \{\mathsf{sk}_i\}_{i\in\mathcal{I}_c})$ を $\mathcal{A}$ に渡す．残りの秘密鍵は保持する．

Step 3. $\mathcal{A}$ は KeyShare オラクルへ任意の多項式回数だけ問い合わせを行う．KeyShare オラクルはクエリ $(i, \mathtt{ID})$ に対して $\mathsf{KeyShare}(\mathsf{mpk}, \mathsf{msk}_i, i, \mathtt{ID})$ を返す．ただし $\mathtt{ID} \neq \mathtt{ID}^*$ に制限される．

Step 4. チャレンジャーは $\mathsf{IBEncap}(\mathsf{mpk}, \mathtt{ID})$ アルゴリズムを実行し，$(c^*, K_1^*)$ を得た後に，$K_0 \xleftarrow{\$} \mathcal{K}_{\mathrm{DEM}}$ を一様ランダムに選ぶ．そして $\beta \xleftarrow{\$} \{0,1\}$ を一様ランダムに選び，$(c^*, K_\beta^*)$ を $\mathcal{A}$ に渡す．ここで $\mathcal{K}_{\mathrm{DEM}}$ は DEM の鍵空間である．

Step 5. $\mathcal{A}$ は任意の多項式回数だけ Step 3 を繰り返す．

Step 6. $\mathcal{A}$ は $\beta' \in \{0,1\}$ を出力する．チャレンジャーは，もし $\beta' = \beta$ ならば 1 を出力し，そうでなければ 0 を出力する．

上記のゲームにおいて，$\beta' = \beta$ であるとき，$\mathcal{A}$ の勝利となる．$\mathcal{A}$ のアドバンテージを

$$Adv^{\mathsf{sID\text{-}TCOA}}_{\mathcal{A},\mathcal{TIBKEM}} := 2 \cdot \Pr[\mathsf{Exp}^{\mathsf{sID\text{-}TCOA}}_{\mathcal{TIBKEM},\mathcal{A}}(\kappa, k, n) = 1] - 1$$

により定義する．

**定義 8.** ある $\kappa_0 \in \mathbb{N}$ が存在して，全ての $\kappa \geq \kappa_0$, $k = \mathrm{poly}(\kappa)$, $n = \mathrm{poly}(\kappa)$ $(k \leq n)$ に対して，全ての確率的多項式時間アルゴリズム $\mathcal{A}$ のアドバンテージ $\mathsf{Adv}^{\mathsf{sID\text{-}TCOA}}_{\mathcal{TIBKEM},\mathcal{A}}(\kappa, k, n)$ が $\kappa$ に関して無視できるほど小さい時，$TIBKEM$ $\mathcal{TIBKEM}$ は IND-sID-TCOA 安全であるという．

[1] で用いられた安全性では，上記に加えて任意の ID をクエリとして入力すると，IBEncap と DecShare の出力である $(K, \{c_i\}_{i\in\mathcal{I}})$ を出力する SessionKey オラクルへのアクセスが許されており，これを IND-sID-CPA 安全性と呼ぶ[*1]．

また，今回定義した安全性は，ゲーム開始時にコラプトするサーバを決定する static corruption モデルになっているが，コラプトするサーバをゲーム開始時に決定せず適応的に決定していくモデルを adaptive corruption モデルという．

## 4. 提案構成法

本節では，TIB-KEM の構成法を示す．本構成法は CBDH 仮定の下で IND-sID-TCOA 安全性を満たすことを示すと共に，IND-sID-CPA 安全性を満たさないことも示す．サーバからの部分復号情報からセッション鍵を再構成するために，以下のラグランジュ係数 $\lambda_i \in \mathbb{Z}_p^*$ を定義する：$|\mathcal{I}| = k$ であるような任意の $\mathcal{I} \subset [n]$ に対し，$\lambda_i = \prod_{j\in\mathcal{I}\setminus\{i\}} \frac{j}{j-i}$．この時，任意の高々 $k-1$ 次の多項式 $Q(P) \in \mathbb{Z}_p[P]$ に対して，$\sum_{i\in\mathcal{I}} Q(i)\lambda_i = Q(0)$ を満たす．

$\mathcal{TIBKEM} = (\mathsf{Init}, \mathsf{KeyShare}, \mathsf{IBEncap}, \mathsf{DecShare}, \mathsf{DecVrfy}, \mathsf{DecCombine})$ を以下のように構成する．

$(\mathsf{mpk}, \mathbf{msk}, \mathbf{vk}) \leftarrow \mathsf{Init}(1^\kappa, k, n)$: $g$ を $\mathbb{G}$ の生成元とし，$R \xleftarrow{\$} \{0,1\}^u$ を選ぶ．$a \xleftarrow{\$} \mathbb{Z}_p$, $h, Y \xleftarrow{\$} \mathbb{G}$ を選び，$X = g^a$ とする．$F : \mathbb{Z}_p \to \mathbb{G}$ となるような関数 $F(I)$ を $F(I) = X^I h$ とする．$\alpha_1, ..., \alpha_{k-1} \xleftarrow{\$} \mathbb{Z}_p$ を選び，多項式 $Q(P) := a + \sum_{i=1}^{k-1} Q_i P^i$ を選ぶ．各 $i \in \mathbb{Z}_p$ に対し，$\mathsf{vk}_i := g^{\alpha(i)}$, $\mathsf{msk}_i := Y^{Q(i)}$ とする．$\mathsf{mpk} = (g, h, X, Y, R)$, $\mathbf{msk} := (\mathsf{msk}_1, \ldots, \mathsf{msk}_n)$, $\mathbf{vk} := (\mathsf{vk}_1, \ldots, \mathsf{vk}_n)$ を出力する．

$\mathsf{sk}[\mathtt{ID}]_i \leftarrow \mathsf{KeyShare}(\mathsf{mpk}, \mathsf{msk}_i, i, \mathtt{ID})$: $s \xleftarrow{\$} \mathbb{Z}_p$ を選び，$\mathsf{sk}_{i,1} := \mathsf{msk}_i \cdot F(I)^s$, $\mathsf{sk}_{i,2} := g^s$ とし，$\mathsf{sk}[\mathtt{ID}]_i := (\mathsf{sk}_{i,1}, \mathsf{sk}_{i,2})$ を出力する．

$(c, K) \leftarrow \mathsf{IBEncap}(\mathsf{mpk}, \mathtt{ID})$: $r \xleftarrow{\$} \mathbb{Z}_p$ を選び，$c_1 := g^r$, $c_2 := F(I)^r$ とする．$K = f_{\mathrm{GL}}\big(e(X, Y)^r, R\big)$ を計算し，$(c := (c_1, c_2), K)$ を出力する．

$(i, c_i)/\perp \leftarrow \mathsf{DecShare}(\mathsf{mpk}, i, \mathtt{ID}, \mathsf{sk}[\mathtt{ID}]_i, c)$: $\mathsf{sk}[\mathtt{ID}]_i = (\mathsf{sk}_{i,1}, \mathsf{sk}_{i,2})$, $c = (c_1, c_2)$ とする．$c_{i,1} := \mathsf{sk}_{i,2}$, $c_{i,2} := \mathsf{sk}_{i,1}$ として，$c_i := (c_{i,1}, C_{i,2})$ を出力する．

[*1] [1] では IND-sID-TKKA 安全性と呼んでいる．

$0/1 \leftarrow \mathsf{DecVrfy}(\mathsf{mpk}, \mathbf{vk}, i, \mathtt{ID}, c_i, c)$: $\mathbf{vk} = (\mathsf{vk}_1, \ldots, \mathsf{vk}_n)$, $c_i = (c_{i,1}, c_{i,2})$, $c = (c_1, c_2)$ とする．$e(g, c_{i,2}) = e\big(\mathsf{vk}_i, Y\big) \cdot e\big(c_{i,1}, F(I)\big)$ であれば 1 を出力し，そうでなければ 0 を出力する．

$K/\perp \leftarrow \mathsf{DecCombine}(\mathsf{mpk}, \mathbf{vk}, \mathtt{ID}, \{c_i\}_{i\in\mathcal{I}}, c)$: $c_i = (c_{i,1}, c_{i,2})$, $c = (c_1, c_2)$ とする．$B_1 := \prod_{i\in\mathcal{I}} c_{i,1}^{\lambda_i}$, $B_2 := \prod_{i\in\mathcal{I}} c_{i,2}^{\lambda_i}$ として．$K = f_{\mathrm{GL}}\Big(\frac{e(c_1,B_2)}{e(c_2,B_1)}, R\Big)$ を計算して出力する．

検証・復号の正当性を以下に示す．

まず検証の正当性について，部分復号情報 $c_i$ が全て正当なものであると仮定すると，

$$e(g, c_{i,2}) = e(g, Y^{Q(i)} \cdot F(I)^r) = e(g, Y^{Q(i)}) \cdot e(g, F(I)^r),$$
$$e(\mathsf{vk}_i, Y) \cdot (c_{i,1}, F(I)) = e(g^{Q(i)}, Y) \cdot e(g^r, F(I))$$
$$= e(g, Y^{Q(i)}) \cdot e(g, F(I)^r).$$

よって $e(g, c_{i,2}) = e\big(\mathsf{vk}_i, Y\big) \cdot e\big(c_{i,1}, F(I)\big)$ が成立し，検証の正当性が示された．

次に復号の正当性を示す．部分復号情報 $\{c_i\}_{i\in\mathcal{I}}$ が全て正当なものであると仮定して，$r' = \sum_{i\in\mathcal{I}} r'_i\lambda_i, s' = \sum_{i\in\mathcal{I}} s_i\lambda_i$ とする．

$$B_1 = \prod_{i\in\mathcal{I}} c_{i,1}^{\lambda_i} = Y^{Q(i)\cdot\lambda_i} \cdot F(I)^{s_i\cdot\lambda_i}$$
$$= Y^{\sum_{i\in\mathcal{I}} Q(i)\lambda_i} \cdot F(I)^s = Y^a \cdot F(I)^s,$$
$$B_2 = \prod_{i\in\mathcal{I}} c_{i,2}^{\lambda_i} = g^{\sum_{i\in\mathcal{I}} s_i\lambda_i} = g^s.$$

従って，

$$\frac{e(c_1,B_2)}{e(c_2,B_1)} = \frac{e(g^r, Y^a \cdot F(I)^s)}{e(F(I)^r, g^s)} = e(g^r, Y^a) \cdot \frac{e(g^r, F(I)^s)}{e(F(I)^r, g^s)}$$
$$= e(g^r, Y^a) = e(g, Y)^{ar} = e(X, Y)^r,$$
$$K = f_{GL}\Big(\frac{e(c_1,B_2)}{e(c_2,B_1)}, R\Big) = f_{GL}\Big(e(X,Y)^r, R\Big).$$

以上よりセッション鍵の復号の正当性が示された．

**定理 1.** CBDH 仮定が成り立つならば，上記のように構成された *TIB-KEM* $\mathcal{TIBKEM}$ は IND-sID-TCOA 安全である．

証明． IND-sID-TCOA ゲームに対する攻撃者を $\mathcal{A}$ とする．IND-sID-TCOA 安全性を破る $\mathcal{A}$ を用いて GL-DBDH 問題を多項式時間で解くことのできるアルゴリズム $\mathcal{B}$ を構築する．アルゴリズム $\mathcal{B}$ にはチャレンジインスタンスとして $(g, A, B, C, L, R)$ が与えられる．$L$ にはランダムに抽出された値 $U$ か $K = f_{\mathrm{GL}}(\mathsf{bdh}(A, B, C), R)$ のどちらかが与えられる．

Step 1. $\mathcal{A}$ はコラプトするサーバのインデックスを任意に選ぶが，ここでは最も情報を得られる場合として $k-1$ 個のインデックスを選ぶものとし，一般性を失わずに $\mathcal{I}_c = \{1, \ldots, k-1\}$ とする．また，同時にターゲット ID を $\mathtt{ID}^*$ も選ぶ．

Step 2. $B$ は以下のように $\mathcal{A}$ との IND-sID-TCOA ゲームを構成する．

(1) $X := A = g^a$, $Y := B = g^b$ とする．$z \xleftarrow{\$} \mathbb{Z}_p$ を選び，$h := X^{-I^*} g^z$ とする．この時，$F(I) := X^I h = X^{I-I^*} g^z$ である．$\mathsf{mpk} := (g, h, X, Y, R)$ として $\mathcal{A}$ に渡す．

(2) コラプトされたサーバ $k-1$ 個分の $\{\mathsf{sk}_i\}_{i\in\mathcal{I}_c}$ を，次のように生成する．$\alpha_1, .., \alpha_{k-1} \xleftarrow{\$} \mathbb{Z}_p$ を選び，全ての $i \in [k-1]$ に対して $\mathsf{sk}_i := Y^{Q(i)}$ とする．この時，$Q(P) \in \mathbb{Z}_p[P]$ を $Q(0) = a$，$Q(i) = \alpha_i$ を満たす $k-1$ 次の多項式とする ($\mathcal{B}$ は $Q(P)$ がわからないことに留意されたい)．$\{\mathsf{sk}_i\}_{i\in\mathcal{I}_c}$ を $\mathcal{A}$ に渡す．

(3) 次のように $\mathbf{vk} := (\mathsf{vk}_1, \ldots, \mathsf{vk}_n)$ を生成する．全ての $i \in [n]$ に対して，$i \in \mathcal{I}_c$ ならば，$vk_i = g^{Q(i)}$ を計算する．$i \notin \mathcal{I}_c$ ならば，$I_i := I_c \cup \{i\}$ として $I_i$ に対しラグランジュ係数 $\lambda_0, \lambda_1, \ldots, \lambda_{k-1} \in \mathbb{Z}_p$ を選ぶ．$Q(i) = \lambda_0 Q(0) + \sum_{j=1}^{k-1} \lambda_j Q(I_j)$ であるから，これを用いて $\mathsf{vk}_i := X^{\lambda_0} \prod_{j=1}^{k-1} \mathsf{vk}_{I_j}^{\lambda_j}$ を計算する．$\mathbf{vk}$ を $\mathcal{A}$ に渡す．

Step 3. $\mathcal{A}$ が任意の $\mathtt{ID}$ $(\neq \mathtt{ID}^*)$, $i \in [n]$ に対して $\mathsf{sk}[\mathtt{ID}]_i$ を求めるクエリを発行したとする．この時，$\mathcal{B}$ は $(i, (\mathsf{sk}_{i,1}, \mathsf{sk}_{i,2}))$ を返さなければならない．$r \xleftarrow{\$} \mathbb{Z}_p$ を選び，$\mathsf{vk}_i$ の生成と同様に，$I_i$ に対しラグランジュ係数 $\lambda_0, \lambda_1, \ldots, \lambda_{k-1} \in \mathbb{Z}_p$ を選ぶ．$Q(i) = \lambda_0 Q(0) + \sum_{j=1}^{k-1} \lambda_j Q(I_j)$ を用いて以下のように導出する．

$$\mathsf{sk}_{i,1} := Y^{\frac{-z\lambda_0}{I-I^*}} F(I)^r \cdot \prod_{j=1}^{k-1} Y^{\lambda_j\alpha_j},$$
$$\mathsf{sk}_{i,2} := Y^{\frac{-\lambda_0}{I-I^*}} g^r. \qquad (1)$$

$\tilde{r} := r - \frac{b\lambda_0}{I-I^*}$ とすると，

$$Y^{\frac{-z\lambda_0}{I-I^*}} F(I)^r = Y^{\frac{-z\lambda_0}{I-I^*}} (X^{I-I^*} g^z)^r$$
$$= \frac{Y^{\frac{-z\lambda_0}{I-I^*}}}{(X^{I-I^*} g^z)^{-\frac{b\lambda_0}{I-I^*}}} (X^{I-I^*} g^z)^{r-\frac{b\lambda_0}{I-I^*}}$$
$$= \frac{1}{X^{-b\lambda_0}} F(I)^{\tilde{r}} = Y^{\lambda_0 a} F(I)^{\tilde{r}}. \qquad (2)$$

(1),(2) 式より，

$$\mathsf{sk}_{i,1} = Y^{\lambda_0 a} F(I)^{\tilde{r}} \cdot \prod_{j=1}^{k-1} Y^{\lambda_j\alpha_j} = Y^{\lambda_0 a + \sum_{j=1}^{k-1} \lambda_j Q(j)} F(I)^{\tilde{r}}$$
$$= Y^{Q(i)} F(I)^{\tilde{r}},$$
$$\mathsf{sk}_{i,2} = g^{\frac{-b\lambda_0}{I-I^*}} g^r = g^{r-\frac{b\lambda_0}{I-I^*}} = g^{\tilde{r}}.$$

$r \xleftarrow{\$} \mathbb{Z}_p$ であるから $\tilde{r} = r - \frac{b\lambda_0}{I-I^*}$ も同様に一様ランダ

ムに分布するため，$\big(i,(\mathsf{sk}_{i,1},\mathsf{sk}_{i,2})\big)$ をクエリに対する有効な返答として $\mathcal{A}$ に返すことができる．

Step 4. $\mathcal{B}$ はチャレンジ暗号文 $c^*=(c_1^*,c_2^*)$ を，以下のように構成する．

$$c_1^* := C,\quad c_2^* := C^z.$$

$K^* := L$ とし，$(c^*,K^*)$ を $\mathcal{A}$ に渡す．

チャレンジ暗号文生成のために $r=c$ とすると，$C=g^c$ であるから $c_2^*=F(I^*)^r=(g^z)^c=C^z$ となり，このチャレンジ暗号文が完全にシミュレートされていることが分かる．

Step 5. $\mathcal{A}$ からのクエリに対し，$B$ は Step 3 と同じように応答する．

Step 6. $\mathcal{A}$ は $b'\in\{0,1\}$ を出力する．$\mathcal{B}$ は $b'$ を受け取り，そのまま出力する．

ここで，攻撃者 $\mathcal{A}$ のアドバンテージ $\mathsf{Adv}^{\mathsf{sID\text{-}TCOA}}_{\mathcal{TIBKEM},\mathcal{A}}$ が無視できないほど大きい時，GL-DBDH 問題を解くアルゴリズム $\mathcal{B}$ のアドバンテージもまた，無視できない大きさとなる．以上より，GL-DBDH 問題を解く確率的多項式時間アルゴリズムが存在しないとき，$\mathcal{TIBKEM}$ は IND-sID-TCOA 安全である．補題 1 より，CBDH 仮定が成り立つならば，$\mathcal{TIBKEM}$ は IND-sID-TCOA 安全である． □

**定理 2.** 上記のように構成された $\mathcal{TIBKEM}$ は IND-sID-CPA 安全性を満たさない．

*Proof.* 上の提案構成法が IND-sID-CPA 安全性を満たさないことを証明する．3 節で述べたように，IND-sID-TCOA ゲームに任意の ID をクエリとして入力すると Encap と DecShare によって出力されるセッション鍵 $K$ と全サーバからの部分復号情報 $c_i$ を返す SessionKey オラクルを追加した．IND-sID-CPA ゲームに勝利するような確率的多項式時間アルゴリズムを持つ攻撃者 $\mathcal{A}$ を構築する．

提案構成法において，DecShare では $(i,\mathtt{ID})$ に対応する秘密鍵をそのまま部分復号情報として出力するため，Step 3 において $\mathcal{A}$ がクエリ $\mathtt{ID}^*$ を発行すると，$\mathtt{ID}^*$ に対応する秘密鍵 $\mathsf{sk}[\mathtt{ID}]$ が手に入る．つまり $\mathtt{ID}^*$ を用いて生成される暗号文 $c^*$ を復号することが可能になるため，$b'=b$ となるような $b'$ を確率 1 で出力する．よって IND-sID-CPA 安全性を満たさない． □

## 5. CBDH 仮定に基づく TPKE 方式

本節では，IND-TCCA 安全な TTKEM とワンタイム安全な DEM から IND-CCA 安全な TPKE が構成できることを示す．その前に，[1] における一般的構成を基にした，TIB-KEM と OTS を用いた TTKEM の構成法を示し，次に IND-CCA 安全な TPKE の構成について示す．

### 5.1 TTKEM：一般的構成法

本節では，TTKEM の一般的構成法について示す．[1] では IND-sID-CPA 安全な TIB-KEM によって IND-TCCA 安全な TTKEM が構成されていたが，ここではさらに弱い IND-sID-TCOA 安全な TIB-KEM を用いて TTKEM の IND-TCCA 安全性を達成している．

$(ek,\mathbf{SK})\leftarrow\mathsf{KeyGen}(1^\kappa,k,n)$：$(\mathsf{mpk},\mathbf{vk},\mathbf{dk})\leftarrow\mathsf{Init}(1^\kappa,k,n)$ を実行し，$\mathsf{ek}:=(\mathsf{mpk},\mathbf{vk})$，$\mathbf{dk}:=\mathbf{msk}$ を出力する．

$(s,K)\leftarrow\mathsf{SessionKeyGen}(\mathsf{ek})$：$(\mathsf{sgk},\mathsf{vrk})\leftarrow\mathsf{KG}(1^\kappa)$ を実行し，$(c,K)\leftarrow\mathsf{IBEncap}(\mathsf{mpk},\mathsf{vrk})$ で暗号化を行う．$s:=(c,\mathsf{vrk},\mathsf{sgk})$ を出力する．

$h\leftarrow\mathsf{Encap}(s,\tau)$：$\sigma\leftarrow\mathsf{Sig}(C||\mathsf{sgk},\tau)$ を実行し，$h:=(c,\sigma,\mathsf{vrk})$ を出力する．

$h_i/\perp\leftarrow\mathsf{PartialDecap}(\mathsf{ek},\mathsf{sk}_i,h,\tau)$：最初に $\mathsf{sk}[\mathsf{vrk}]_i\leftarrow\mathsf{KeyShare}(\mathsf{mpk},\mathsf{msk}_i,\mathsf{vrk},i)$ を実行し，$h_i,/\perp\leftarrow\mathsf{DecShare}(\mathsf{mpk},i,\mathsf{vrk},\mathsf{sk}[\mathsf{vrk}],h)$ を実行，復号結果を出力する．

$K,/\perp\leftarrow\mathsf{Reconst}(\mathsf{ek},h,\{h_i\}_{i\in\mathcal{I}_r},\tau)$：$\mathsf{Vrfy}(C||\tau,\sigma,\mathsf{vrk})=1$ かつ $\mathsf{DecVrfy}(\mathsf{mpk},\mathbf{vk},\mathsf{vrk},i,h_i,c)=1$ ならば，$K\leftarrow\mathsf{DecCombine}(\mathsf{mpk},\mathbf{vk},\mathsf{vrk},\{h_i\}_{i\in\mathcal{I}},C)$ を出力する．

**定理 3.** $\mathcal{TIBKEM}$ が IND-sID-TCOA 安全であり，$\mathcal{OTS}$ が sUF-CMA 安全であれば，上で構成された $\mathit{TTKEM}$ $\mathcal{TTKEM}$ は IND-TCCA 安全である．

紙面の都合上証明は省略するが，[1] と同様に証明が可能である．

### 5.2 TPKE：一般的構成法

本節では，[2] で示されている IND-TCCA 安全な TTKEM とワンタイム安全な DEM を用いた TPKE の一般的構成法を示す[*2]．

$(\mathsf{pk},\mathbf{sk})\leftarrow\mathsf{Setup}(1^\kappa,k,n)$：$(\mathsf{ek},\mathbf{dk})\leftarrow\mathsf{KeyGen}(1^\kappa,k,n)$ を実行し，$\mathsf{pk}:=\mathsf{ek}$，$\mathbf{sk}:=\mathbf{dk}$ として出力する．

$c\leftarrow\mathsf{Enc}(\mathsf{pk},m)$：$(s,K)\leftarrow\mathsf{SessionKeyGen}(\mathsf{ek})$ でセッション鍵を生成し，DEM の暗号化アルゴリズム $c_m\leftarrow\mathsf{E}(K,m)$ を実行する．$h\leftarrow\mathsf{Encap}(s,c_m)$ を実行し，$c:=(c_m,h)$ として出力．

$h_i/\perp\leftarrow\mathsf{PartialDec}(\mathsf{pk},\mathsf{sk}_i,i,c)$：$h_i/\perp\leftarrow\mathsf{PartialDecap}(\mathsf{ek},\mathsf{sk}_i,h,c_m)$ を実行し，復号結果を出力する．

$M/\perp\leftarrow\mathsf{Combine}(\mathsf{pk},c,\{h_i\}_{i\in\mathcal{I}})$：$K/\perp\leftarrow\mathsf{Reconst}(\mathsf{ek},h,\{h_i\}_{i\in\mathcal{I}},c_m)$ を実行し，DEM の復号アルゴリズム $m\leftarrow\mathsf{D}(K,c_m)$ の復号結果を出力する．

**定理 4.** $\mathcal{TTKEM}$ が IND-TCCA 安全であり，$\mathcal{DEM}$ が IND-OT 安全であれば，上で構成された $\mathit{TPKE}$ $\mathcal{TPKE}$ は IND-CCA 安全である．

[*2] 紙面の都合上 DEM の定義は省略する．

| | 結託モデル | 公開鍵長 | 秘密鍵長 | 暗号文長 | 平文空間 | 計算量仮定 |
|---|---|---|---|---|---|---|
| DK05 [11] | adaptive | $\mathcal{O}(n\|\mathbb{G}\|)$ | $\mathcal{O}(\|\mathbb{G}\|)$ | $\mathcal{O}(n\|\mathbb{G}_T\|)$ | $\{0,1\}$ | CDH |
| BBH06 [4] | static | $4\|\mathbb{G}\| + \|\mathsf{vk}\|$ | $2\|\mathbb{G}\|$ | $3\|\mathbb{G}_T\| + \|\mathsf{vrk}\| + \|sig\|$ | $\mathbb{G}_T$ | DBDH |
| 提案構成法 | static | $5\|\mathbb{G}\| + \|\mathsf{vk}\|$ | $2\|\mathbb{G}\|$ | $2\|\mathbb{G}_T\| + \|\mathsf{vrk}\| + \|sig\| + \|c_m\|$ | $\{0,1\}$ | CBDH |

**表 1** TPKE 構成法の比較．$|\mathbb{G}|, |\mathbb{G}_T|$ はそれぞれ $\mathbb{G}$ と $\mathbb{G}_T$ の要素のビット長を表す．また，$|\cdot|$ は各要素のビット長を表すものとする．

## 6. 構成法の比較

提案構成法は，公開鍵や秘密鍵，暗号文が定数倍長となるような構成になっている．表 5.1 にいくつかの既存方式 [4], [11] と我々の提案構成法の効率性の比較を示す．具体的には，結託モデル，鍵長，暗号文長，計算量仮定に関して比較を行なっている．[11] では ME に関して，PKE を用いた一般的構成と 2-level selective Hierarhical IBE (HIBE) を用いる一般的構成の二種類が提案されており，従って得られる TPKE の構成法も二種類となるが，簡単のため前者との比較を行う．具体的には，[13] に基づく CDH 仮定に基づく PKE を用いたときのパラメータと比較を行う．各サーバに対して PKE の鍵ペアを用意するため，公開鍵長及び暗号文長が $n$ に依存している．DBDH 仮定に基づく TPKE[4] と比較すると，提案方式の方が平文空間が制限されているものの，同程度のパラメータを達成していることがわかる．

## 7. まとめ

本稿では，TIB-KEM における弱い安全性として IND-sID-TCOA 安全性を新たに定義し，その安全性を満たす方式を CBDH 仮定から構成できることを示した．[1] の結果を用い，IND-sID-TCOA 安全な TIBKEM とワンタイム署名を組み合わせることで IND-TCCA 安全な TTKEM を構成し，更にワンタイム安全な DEM と組み合わせることで，初めて CCA 安全かつパラメータ長が効率的な TPKE 方式を CBDH 問題のような計算問題の困難性に基づいて構成することができた．

**謝辞** 本研究は JSPS 科研費 18H03238 の助成を受けたものです．

### 参考文献

[1] 三田 隼平, 四方 順司, "スタンダードモデルにおける Threshold Tag-KEM の一般的構成法", CSS2013.

[2] Masayuki Abe, Rosario Gennaro, Kaoru Kurosawa, and Victor Shoup. "Tag-KEM/DEM: A new framework for hybrid encryption and a new analysis of kurosawa-desmedt KEM." In Ronald Cramer, editor, EUROCRYPT 2005, volume 3494 of LNCS, pages 128-146, Aarhus, Denmark, May 22-26, 2005.

[3] Dan Boneh and Xavier Boyen. "Secure identity based encryption without random oracles." In Matt Franklin, editor, Advances in Cryptology-CRYPTO 2004, volume 3152 of LNCS, pages 443-59. Springer-Verlag, 2004.

[4] Dan Boneh, Xavier Boyen, and Shai Halevi. "Chosen ciphertext secure public key threshold encryption without random oracles." In Topics in Cryptology-CT-RSA 2006, volume 3860 of Lecture Notes in Computer Science, pages 226-243. Berlin: Springer-Verlag, 2006.

[5] D. Boneh, R. Canetti, S. Halevi, J. Katz, "Chosen-Ciphertext Security from Identity-Based Encryption." SIAM J. of Computing, 36(5), pp. 1301-1328, 2007.

[6] Xavier Boyen, Qixiang Mei, and Brent Waters. "Simple and efficient CCA2 security from IBE techniques." In ACM CCS 2005, pages 320-329. New-York: ACM Press, 2005.

[7] R. Canetti and S. Goldwasser. "An efficient threshold public key cryptosystem secure against adaptive chosen ciphertext attack." In Eurocrypt'99, pages 90-106, 1999.

[8] Ran Canetti, Shai Halevi, and Jonathan Katz. "Chosen-ciphertext security from identity based encryption." In Advances in Cryptology - EUROCRYPT 2004, volume 3027 of LNCS, pages 207-222, 2004.

[9] Ronald Cramer and Victor Shoup. "A practical public key cryptosystem provably secure against adaptive chosen ciphertext attacks." In: Crypto 1998, volume 1462 of LNCS, pages 13-25. Springer-Verlag, 1998.

[10] Ronald Cramer and Victor Shoup. "Universal hash proofs and a paradigm for adaptive chosen ciphertext secure public-key encryption." In: EUROCRYPT 2002, volume 2729 of LNCS, pages 45-64, 2002.

[11] Y. Dodis and J. Katz. "Chosen-ciphertext security of multiple encryption." InTCC' 05, volume 3378 ofLNCS, pages 188-209. Springer, 2005.

[12] Oded Goldreich and Leonid A. Levin. "A hard-core predicate for all one-way functions." In Proceedings of the Twenty-First Annual ACM Symposium on Theory of Computing, STOC, pages 25-32. ACM, 1989.

[13] Kristiyan Haralambiev, Tibor Jager, Eike Kiltz, and Victor Shoup. "Simple and efficient public-key encryption from computational diffie-hellman in the standard model." In Public Key Cryptography - PKC 2010, volume 6056 of LNCS, pages 1-18. Springer, 2010.

[14] T. Ishihara, H. Aono, S. Hongo, J. Shikata, "Construction of Threshold (Hybrid) Encryption in the Random Oracle Model: How to Construct Secure Threshold tag-KEM from Weakly Secure Threshold KEM." Information Security and Privacy, LNCS 4586, pp. 259-273, Springer, 2007.

[15] E. Kiltz, D. Galindo, "Threshold chosen-ciphertext secure identity-based key encapsulation without random oracles." In SCN 2006, volume 4116 of LNCS, pages 173-185. Springer-Verlag, 2006.

[16] Chen Yu, Liqun Chen, and Zongyang Zhang. " CCA secure IB-KEM from the Computational Bilinear Diffie-Hellman Assumption in the Standard Model." Information Security and Cryptology-ICISC 2011. Springer Berlin Heidelberg, 2012. 275-301.