修士論文

時間前開封機能付き時限式暗号に関する研究

A Study on Timed-Release Encryption with Pre-open Capability

指導教員　　松浦幹太　准教授

東京大学大学院 情報理工学系研究科 電子情報学専攻

86420　　中井 泰雅

平成22年2月9日提出

内容梗概

近年，インターネットの急速な普及により，情報セキュリティの重要性が高まってきている．そして，通信の秘匿性を確保することや，情報の改竄を防ぐということは，情報セキュリティにおける重要な課題である．暗号技術は，これらの課題を解決するための中心的役割を果たす通信の基礎技術である．暗号を利用する際には，鍵配送が大きな問題となるが，それを実現する手法として，公開鍵暗号 (Public Key Encryptin, PKE) がある．この方式は，事前の秘密情報（鍵情報）の共有無しで安全な秘匿通信を行うための技術であり，実社会においてSSLなどのプロトコルの根幹を支えるものとして欠くことができない．

しかし，公開鍵暗号においては，公開鍵と秘密鍵の正当性の保証が必要となり，その管理が大きな負荷となる．その問題を解決するために提唱されたのがIDベース暗号方式 (Identity Based Encryption, IBE) である．この方式を用いれば，各人のID（メールアドレス等）をもとにした公開鍵を用いるため，鍵の管理が容易となる．

それ以降，様々な機能をもった暗号方式が提案されてきた．本研究ではその中に一つである時間前開封機能付き時限式暗号 (Timed-Release Encryption with Pre-open Capability, TRE-PC) についての研究を行っている．TRE-PCとは，送信者の指定した時刻になるか，もしくは時間前開封鍵を送信者が受信者に送信するまで復号することのできない暗号である．

TRE-PCが提案されて以降，いくつかのTRE-PCの構成法が提案されてきた．しかし，それらの方式はすべて具体的な数論的問題の困難性に基づいて安全性の証明がされているものであり，一般的な方式ではなかった．

そこで本研究では，TRE-PCの一般的構成法を示すことを目指した．一般的構成法とは，既存の一般的に知られている暗号技術を構成要素として，異なる暗号技術を構成する手法であり，実装の際に非常に役に立つと考えられる．まず，(時間前開封機能の付かない) 普通の時限式暗号 (TRE) の既に知られている一般的構成法と本質的に同じ構成でTRE-PCを一般的に構成することができることを示した後，さらに暗号文サイズの観点からより効率的にTRE-PCを一般的に構成できることを示した．それぞれの方式について，TRE-PCで求められると考えられる安全性についての証明も行っている．

目次

Chapter 1　序論

1.1　暗号方式について

近年，インターネットの急速な普及により，情報セキュリティの重要性が高まってきている．そして，通信の秘匿性を確保することや，情報の改竄を防ぐということは，情報セキュリティにおける重要な課題である．暗号技術は，これらの課題を解決するための中心的役割を果たす通信の基礎技術である．暗号を利用する際には，鍵配送が大きな問題となるが，それを実現する手法として，公開鍵暗号 (Public Key Encryptin, PKE) がある [29]．この方式は，事前の秘密情報（鍵情報）の共有無しで安全な秘匿通信を行うための技術であり，実社会において SSL などのプロトコルの根幹を支えるものとして欠くことができない．

公開鍵暗号とは，暗号化用の鍵と復号用の鍵が異なっていて，暗号化用の鍵から復号用の鍵を求めることが計算量的に困難であるものを指す．復号用の鍵を秘密にしておく限り，暗号化用の鍵は公開しても安全な通信ができる．このため通常，暗号化用の鍵を公開鍵，復号用の鍵を秘密鍵と呼ぶ．

公開鍵暗号は 1976 年の Diffie と Hellman [29] の概念の提唱以降，その構成法や安全性の議論が活発に行われてきた．最も有名な公開鍵暗号方式の一つは RSA 暗号 [46] であろう．

それ以降，公開鍵暗号では様々な種類のものが提案されてきた．例えば，公開鍵と秘密鍵の正当性の保証の管理にかかる負荷を解決するために提唱された，ID ベース暗号方式 (Identity Based Encryption, IBE)[47] や，公開鍵に証明書をつけずに済むように考えられた Certificateless Encryption[5] といったものがある．本研究で考えている時間前開封機能付き時限式暗号 (Timed-Release Encryption with Pre-open Capability, TRE-PC) もそのような機能付き暗号の一つである．詳しくは 3 章を参照されたい．

1.2　証明可能安全性

あらゆる暗号技術は，現在知られている困難な問題に基づくなどして，安全性を証明可能であることが望ましい．証明可能安全性とは，暗号の安全性を形式的に定義するためのものであり，帰着する数学的証明の正当性によって，定義の範囲内の安全の有無を判断できるようにするものである．安全性の証明がないからといって必ずしも安全ではないというわけではないが，より正確で，客観的な安全性の議論を行うために，新たな方式考案の際など，安全性の証明をつけることは事実上必ず必要なものとなっている．

証明可能安全性を示すためには，第一に，示したい安全性目標のモデル，攻撃者の攻撃法のモデル，根拠とする困難な問題の形式的な定義を行う必要がある．根拠とな

る問題は，例えば素因数分解問題や離散対数問題など，長くにわたって困難であると信じられている問題を使うことが多い．また，一般的構成法などの場合では，構成する暗号方式 (公開鍵暗号や ID ベース暗号など) の安全性を，根拠となる問題として用いることが多くなっている．

安全性の定義は “現実に知られている難しい問題の困難性の仮定が成り立つならば，安全性を無視できない確率で破るアルゴリズムが存在しない” というものになっている．証明の際には，その対偶を示すことで行われる．つまり，“安全性を無視できない確率で破るアルゴリズムが存在するならば，現実に知られている難しい問題の困難性を破ることができる” という証明を行うことがほとんどである．

具体的には，安全性の証明をしたい方式を破る確率多項式時間アルゴリズムを，入出力の決まったブラックボックスとして考えた上で，その攻撃者を利用すると，安全性の根拠としたい困難な問題を解くことができるアルゴリズム[1]の存在を示すことができ，問題の困難性の仮定を破ることから，対偶により安全性を破る攻撃者は存在しないとするのである．確率多項式時間アルゴリズムは，現実に存在するアルゴリズムの能力を表している．この場合は方式の安全性を決定するセキュリティパラメータに対して多項式時間である．

ランダムオラクルモデルとスタンダードモデル　証明可能安全性の枠組みにおいて，証明を考えるときにランダムオラクルモデル (Random Oracle Model) [30, 8] と呼ばれるモデルが使用されることがある．ランダムオラクルモデルとは，誰でもアクセスできる真にランダムな関数 (ランダムオラクル) が存在すると仮定するモデルのことである．これに対し，ランダムオラクルを使用しないモデルをスタンダードモデル (Standard Model) という．ランダムオラクルは，出力が真にランダムな値で，また，出力空間において一様分布であるとみなせるハッシュ関数であるとも考えることができる．実用の際には，ランダムオラクルの代わりに SHA-1 や SHA-256 などの実用的なハッシュ関数を使用する．一般的に，ランダムオラクルを使う方式の方が，スタンダードモデルでの方式よりも計算コストや暗号文サイズなどの面で効率のよいものができる．実際に使用されている RSA [1]，あるいは RSA-PSS [9] などといった多くの方式がランダムオラクルモデルでのものである

便利なツールであるランダムオラクルだが，あくまで理想的な存在であり，現実世界においては，ランダムオラクルのような真にランダムな出力を持つ関数は存在しない．また，ランダムオラクルモデルにおいて証明可能安全性を有する方式は，そのランダムオラクル以外のどのような関数に置き換えてもスタンダードモデルでは安全性を証明できなくなるものも多く存在することが分かっている [15, 43, 32, 7].

このような理由により，最初にランダムオラクルモデルで設計して安全性を証明し，後に改良を加えてスタンダードモデルで証明できる方式を考えるということは行われることはあるものの，電子署名に限らず，新しく暗号学的な方式を考える際は，ラ

[1]攻撃者にとっては，安全性の定義に用いられるゲーム (2 章や 3 章参照) でのやりとりをしているように見せられるということを示さなければならない．攻撃者とチャレンジャーのやりとりをシミュレートすることから，このようなアルゴリズムをシミュレータ，あるいは，帰着アルゴリズム (Reduction Algorithm) という．本稿では前者の呼び名を用いる．

ンダムオラクルを使わなくてもよいならば，ランダムオラクルを用いない方式を目指す研究が増えている．

一方で，実際に設計される場合にはランダムオラクルモデルが用いられる場合が多い．なぜなら，スタンダードモデルで実装するためには，先にも述べたように暗号文サイズや計算コストの面で，実用的でない場合が多いからである．また，ランダムオラクルモデルで証明されたものは確かに，スタンダードモデルでは安全性を証明できていないものもあるのだが，実用的な暗号技術の中で，ランダムオラクルだから破られた，という例が報告されていないため，実用上ランダムオラクルモデルでも問題がない場合がほとんどであるという理由もある．

上記の理由から，実用の際には，用途や必要となる安全性のレベルによって，それぞれのモデルの方式をうまく使い分けることが必要になるのではないかと考えられる．つまり，どうしても速さなどが欲しい場合にはランダムオラクルモデルの方式を，絶対に守らなければならない用途に使用する場合にはスタンダードモデルの方式を使用する，といった使い分けが必要になると考えられる．

1.3　一般的構成法

一般的構成法とは，既存の一般的に広まっていると考えられる暗号方式を構成要素として用いて，別の暗号方式を構成する，という手法である．1.2 で言う安全性の根拠を，具体的な数論の困難性 (素因数分解問題や離散対数問題) ではなく，構成要素となる暗号方式をブラックボックスとして用いて，それぞれの安全性要件を，構成する暗号方式の安全性の根拠として用いる．つまり，既に安全性要件を満たしている構成要素となる暗号方式さえ存在していれば，それらを組み合わせることで新たな暗号方式を構成することが可能であることになる．

多くの場合，一般的構成法で構成した方式は，具体的な数論の困難性に基づいて構成された方式に比べて計算コストや暗号文サイズが大きくなる．なぜなら，一般的構成法では，既存の暗号方式から新しい暗号方式を構成する際に，安全性要件を満たすためにいくつかの工夫をし，具体的な数論の困難性に基づく場合に比べて遠回りをして構成する必要があることがほとんどだからである．

一方で，一般的構成法が存在することで，新しい暗号方式を構成しやすいという利点がある．具体的な数論の困難性に基づいて新しく暗号方式を構成しようとすると，実装の際に一から構成を行っていく必要があるため，実装コストが非常に大きい．しかし，一般的構成法が存在していれば，構成要素となる暗号方式さえ自身で所持していれば，それらを正しく組み合わせるだけで新しい暗号方式を安全に構成することが可能となるため，実装コストを非常に小さく抑えられる可能性がある．このことは，実際に運用することまで考えると，非常に大きな利点となりうる．

1.4　本研究の貢献

本研究では，時間前開封機能付き時限式暗号 (TRE-PC) の一般的構成法を初めて提案した．これは，構成要素となる暗号方式がスタンダードモデルであれば，スタンダードモデルで安全性証明を行うことができる．また，その後この方式を改良し，暗号文オーバーヘッドを小さくしたものも提案している．2 種類の方式を提案しており，片方は初期のものと同様に，構成要素がスタンダードモデルで安全性証明を行うことができるものであり，もう一方はランダムオラクルモデルでの安全性ではあるものの非常に効率のよい方式となっている．

提案方式の 1 つ目は公開鍵暗号，ID ベース暗号，および，電子署名から成る方式であり，2 つ目は，鍵カプセル化メカニズム，ID ベース鍵カプセル化メカニズム，Encapsulation，共通鍵暗号，および，メッセージ認証子から成る方式，3 つ目は，鍵カプセル化アルゴリズム，および ID ベース鍵カプセル化アルゴリズムから成る方式となっている．

1.5　本稿の構成

以下，2 章では，3 章以降の説明の際に必要となる暗号方式やその安全性などの諸定義を説明する．3 章では，本研究で構成する時間前開封機能付き時限式暗号，およびその元となった時限式暗号の方式と，その安全性証明を示す．4 章では，本研究の関連研究を紹介する．5 章では，提案方式と，それぞれの提案方式の安全性証明を行う．6 章では，本研究で提案した手法に関するいくつかの議論を行う．7 章は本稿のまとめである．

なお，5 章の 5.2 節の提案方式を中心とする本稿の内容の一部は，査読無し国内会議 SCIS 2009(発表文献 [ii])，査読付き国際会議 IWSEC 2009(発表文献 [i]) において発表した．また，5 章の 5.3 節，5.4 節の提案方式を中心とする本稿の内容の一部は，査読無し国内会議 SCIS 2010(発表文献 [iii]) において発表した．

Chapter 2　諸定義

本章では，3 章以降で必要となる様々な暗号技術のアルゴリズムとその安全性について，各々の概要と定義を振り返る．

本稿での記号の定義　本稿では，$x \leftarrow y$ と書くとき，y が集合ならばそこから一様ランダムに要素を取り出し x に代入する操作を，y がアルゴリズムまたは関数ならば x を出力する操作を表す．“$x||y$” は x と y の連結を表す．また，$\mathcal{A}$ が確率的アルゴリズムであり，$y \leftarrow \mathcal{A}(x;r)$ と書くとき，$\mathcal{A}$ は x を入力，r を乱数として用いて計算し，y を出力することを意味する．また，$\Pr[x]$ とは，x が起こる確率を表すこととする．

2.1　公開鍵暗号 (PKE)

公開鍵暗号 Π は以下の 3 つの確率的アルゴリズムからなる．

鍵生成 PKE.KG: セキュリティパラメータ 1^κ を入力とし，秘密鍵 sk と公開鍵 pk の対を出力する．

暗号化 PKE.Enc: 公開鍵 pk，平文 $m \in \mathcal{M}$ を入力とし，暗号文 $c \in \mathcal{C}$ を出力する．

復号 PKE.Dec: 秘密鍵 sk，暗号文 c を入力とし，平文 m (あるいは $\perp$) を出力する．

$\mathcal{M}$，$\mathcal{C}$ はそれぞれ，Π の平文空間，暗号文空間である．

正当性　PKE.KG から出力された全ての鍵ペア (sk, pk)，全ての $m \in \mathcal{M}$ に対し，以下を満たさなければならない．

$$\mathsf{PKE.Dec}(sk, \mathsf{PKE.Enc}(pk, m)) = m$$

2.1.1　IND-CCA 安全性

公開鍵暗号の適応的選択暗号文攻撃に対する識別不可能性 (Indistinguishability against Adaptive Chosen-Ciphertext Attacks, IND-CCA) は，以下の攻撃者 $\mathcal{A}$ と IND-CCA Challenger $\mathcal{CH}$ 間の IND-CCA game を用いて定義される．この定義は，例え攻撃対象以外の任意の暗号文に対して復号してもらうことが可能な環境にある攻撃者に対してであっても，暗号文から平文の情報が 1bit も漏れない，という安全性の定義になっている．

Setup. $\mathcal{CH}$ は $\mathsf{PKE.KG}(1^\kappa)$ を実行する．出力された pk を $\mathcal{A}$ に渡し，sk を保持しておく．

Phase 1. $\mathcal{A}$ は $\mathcal{CH}$ に対し，任意の回数，復号クエリ c を発行することができる．$\mathcal{CH}$ は c に対し，正しい復号結果 $m \leftarrow \mathsf{PKE.Dec}(sk, c)$ $(m \in \mathcal{M} \cup \perp)$ を返す．

Challenge. $\mathcal{A}$ は 2 つの任意の平文 m_0, m_1 を選び，$\mathcal{CH}$ に送る．$\mathcal{CH}$ はランダムにコイン $b_C \in \{0, 1\}$ を振り，m_{b_C} の暗号文 $c^* \leftarrow \mathsf{PKE.Enc}(pk, m_{b_C})$ を計算し，c^* を $\mathcal{A}$ に渡す．

Phase 2. $\mathcal{A}$ は Phase 1 と同様に復号クエリを発行することができる．ただし，Challenge の際受け取った暗号文 c^* を復号クエリとすることはできない．

Guess. $\mathcal{A}$ は $\mathcal{CH}$ の選んだ b_C の予測として b_A を出力する．

ここで，ある公開鍵暗号 Π における $\mathcal{A}$ の IND-CCA アドバンテージを以下の様に定義する．

$$\mathsf{Adv}^{\text{IND-CCA}}_{\Pi,\mathcal{A}} = |\Pr[b_A = b_C] - 1/2|$$

定義 2.1. 全ての多項式時間アルゴリズム $\mathcal{A}$ に対し，$\mathsf{Adv}^{\text{IND-CCA}}_{\Pi,\mathcal{A}}$ が無視できるほど小さい時，Π は *IND-CCA* 安全であるという．

2.2　ID ベース暗号 (IBE)

公開鍵暗号における鍵のすり替え問題を解決するために，公開鍵の正当性を保証しなければならない．Shamir はその問題を解決するために各エンティティの識別子を公開鍵として利用する，ID ベース暗号 (IBE) という概念を 1984 年に提案した [47]．ID ベース暗号 Π は以下の 4 つの確率的アルゴリズムからなる．

セットアップ IBE.Setup: セキュリティパラメータ 1^κ を入力とし，マスター秘密鍵 msk と公開パラメータ prm の対を出力する．

鍵導出 IBE.Ext: 公開パラメータ prm，マスター秘密鍵 msk，$\mathsf{ID} \in \mathcal{I}$ を入力とし，ID に対する復号鍵 d_{ID} を出力する．

暗号化 IBE.Enc: 公開パラメータ prm，$\mathsf{ID} \in \mathcal{I}$，平文 $m \in \mathcal{M}$ を入力とし，暗号文 $c \in \mathcal{C}$ を出力する．

復号 IBE.Dec: 復号鍵 d_{ID}，暗号文 c を入力とし，平文 m (あるいは $\perp$) を出力する．

$\mathcal{I}$，$\mathcal{M}$，$\mathcal{C}$ はそれぞれ，Π の ID 空間，平文空間，暗号文空間である．

正当性 IBE.Setup から出力された全ての $(\mathsf{msk}, \mathsf{prm})$,$\mathsf{ID} \in \mathcal{I}$, $d_{\mathsf{ID}} \leftarrow \mathsf{IBE.Ext}(\mathsf{prm}, \mathsf{msk}, \mathsf{ID})$，および全ての $m \in \mathcal{M}$ に対し，以下を満たさなければならない．

$$\mathsf{IBE.Dec}(d_{\mathsf{ID}}, \mathsf{IBE.Enc}(\mathsf{prm}, \mathsf{ID}, m)) = m$$

2.2.1　IND-ID-CPA 安全性

IBE の適応的 ID および選択平文攻撃に対する識別不可能性 (Indistinguishability against Adaptive ID and Adaptive Chosen-Plaintext Attacks, IND-ID-CPA) は，以下の攻撃者 $\mathcal{A}$ と IND-ID-CPA Challenger $\mathcal{CH}$ 間の IND-ID-CPA game を用いて定義される．この定義は，例え任意の平文に対する暗号文を得ることが可能な環境にある攻撃者に対してであっても，暗号文から平文の情報が 1bit も漏れない，という安全性の定義になっている．

Setup. $\mathcal{CH}$ は $\mathsf{IBE.Setup}(1^\kappa)$ を実行する．出力された prm を $\mathcal{A}$ に渡し，msk を保持しておく．

Phase 1. $\mathcal{A}$ は $\mathcal{CH}$ に対し，任意の回数，鍵導出クエリ ID を発行することができる．$\mathcal{CH}$ はそれぞれのクエリ ID に対し，正しい復号鍵 $d_{\mathsf{ID}} \leftarrow \mathsf{IBE.Ext}(\mathsf{prm}, \mathsf{msk}, \mathsf{ID})$ を返す．

Challenge. $\mathcal{A}$ は 2 つの任意の平文 m_0, m_1，および鍵導出クエリとして発行していない ID^* を選び，$\mathcal{CH}$ に送る．$\mathcal{CH}$ はランダムにコイン $b_C \in \{0,1\}$ を振り，m_{b_C} の暗号文 $c^* \leftarrow \mathsf{IBE.Enc}(\mathsf{prm}, \mathsf{ID}^*, m_{b_C})$ を計算し，c^* を $\mathcal{A}$ に渡す．

Phase 2. $\mathcal{A}$ は Phase 1 と同様に鍵導出クエリを発行することができる．ただし，$\mathcal{A}$ は ID^* を鍵導出クエリとすることはできない．

Guess. $\mathcal{A}$ は $\mathcal{CH}$ の選んだ b_C の予測として b_A を出力する．

ここで，ある ID ベース暗号 Π における $\mathcal{A}$ の IND-ID-CPA アドバンテージを以下の様に定義する．

$$\mathsf{Adv}^{\text{IND-ID-CPA}}_{\Pi,\mathcal{A}} = |\Pr[b_A = b_C] - 1/2|$$

定義 2.2. 全ての多項式時間アルゴリズム $\mathcal{A}$ に対し，$\mathsf{Adv}^{\text{IND-ID-CPA}}_{\Pi,\mathcal{A}}$ が無視できるほど小さいとき，Π は *IND-ID-CPA* 安全であるという．

2.2.2　乱数に関するターゲット衝突困難性

提案手法の安全性証明に必要となる乱数に関するターゲット衝突困難性を導入する．簡単に言うと，乱数 R_{IBE} や暗号化アルゴリズム $\mathsf{IBE.Enc}$，およびマスター秘密鍵 msk さえ与えられているいかなる攻撃者であっても，平文，ID，および与えられたものと異なる乱数 R'_{IBE} を用いて暗号化した結果が R_{IBE} を用いて暗号化した結果と衝突するような R'_{IBE} を見つけることができない，という安全性である．「ターゲット衝突困難性」という言葉を用いる理由は，ハッシュ関数に対するターゲット衝突困難性 [42, 10] と同様に，攻撃者は乱数のうちの一つに対する制御能力を持っていないからである．ここで注意したいのは，[12, 49, 31] のように，多くの実用的な IBE 方式は，何の仮定もなしにこの安全性を満たすということである．

以下の攻撃者 $\mathcal{A}$ と Challenger $\mathcal{CH}$ 間のゲームで定義する．

$$\mathsf{Adv}^{\mathrm{Rand}}_{\Pi,\mathcal{A}} = \Pr\left[\begin{array}{l}(\mathsf{msk},\mathsf{prm}) \leftarrow \mathsf{IBE.Setup}(1^\kappa); R_{IBE} \leftarrow \mathcal{R}_{IBE};\\ (m', \mathsf{ID}', R'_{IBE}) \leftarrow \mathcal{A}(\mathsf{msk},\mathsf{prm}, R_{IBE}) :\\ \mathsf{IBE.Enc}(\mathsf{prm},\mathsf{ID}^*, m'; R'_{IBE}) =\\ \mathsf{IBE.Enc}(\mathsf{prm},\mathsf{ID}^*, m'; R_{IBE}) \wedge R'_{IBE} \neq R_{IBE}\end{array}\right]$$

$\mathcal{R}_{IBE}$ は暗号化アルゴリズムの乱数空間を表す．

定義 2.3. 全ての多項式時間アルゴリズム $\mathcal{A}$ に対し，$\mathsf{Adv}^{\mathrm{Rand}}_{\Pi,\mathcal{A}}$ が無視できるほど小さいとき，Π は乱数に関するターゲット衝突困難性を持つという．

2.3　One-Time 署名

One-Time 署名 Σ は以下の 3 つのアルゴリズムから成る．

鍵生成 SigKG: セキュリティパラメータ 1^κ を入力とし，署名鍵 SK と検証鍵 VK の対を出力する．

署名 Sign: 署名鍵 SK，平文 $m \in \mathcal{M}$ を入力とし，署名 σ を出力する．

検証 Verify: 検証鍵 VK，平文 m，署名 σ を入力とし，accept/reject を返す．

$\mathcal{M}$ は Π の平文空間である．

正当性　SigKG から出力された全ての鍵ペア (VK, SK)，および全ての $m \in \mathcal{M}$ に対し，以下を満たさなければならない．

$$\mathsf{Verify}(VK, \mathsf{Sign}(SK, m)) = \mathtt{accept}$$

2.3.1　SUF-OTsig 安全

One-Time 署名安全性の定義として，強偽造不可性 (Strong Unforgeability) がある．以下の攻撃者 $\mathcal{A}$ と SUF-OTsig Challenger $\mathcal{CH}$ 間の SUF-OTsig game を用いて定義される．この定義は，例え高々1 回のみ，平文に対する署名のクエリを発行することができる攻撃者に対してであっても，発行した (m, σ) ではなく，かつ Verify が accept となるような平文と署名の組を偽造することはできない，という安全性の定義となっている．

Setup. $\mathcal{CH}$ は $\mathsf{SigKG}(1^\kappa)$ を実行する．出力された VK を $\mathcal{A}$ に渡し，SK を保持しておく．

Query. $\mathcal{A}$ は $\mathcal{CH}$ に対し，高々1 回のみの署名クエリ m を発行することができる．$\mathcal{CH}$ は m に対し，正しい署名 $\sigma \leftarrow \mathsf{Sign}(SK, m)$ を返す．

Forge. $\mathcal{A}$ は平文と署名の組 (m^*, σ^*) を出力する

ここで，ある電子署名 Π のアドバンテージを以下のように定義する．

$$\mathsf{Adv}^{\text{SUF-OTsig}}_{\Sigma,\mathcal{A}} = \Pr[\mathsf{Verify}(VK, \sigma^*) = \mathtt{accept} \wedge (m, \sigma) \neq (m^*, \sigma^*)]$$

定義 2.4. 全ての多項式時間アルゴリズム $\mathcal{A}$ に対し，$\mathsf{Adv}^{\text{SUF-OTsig}}_{\Sigma,\mathcal{A}}$ が無視できるほど小さい時，Σ は *SUF-OTsig* 安全という．

2.4　KEM/DEM フレームワーク

KEM(Key Encapsulation Mechanism)/DEM(Data Encapsulation Mechanism) フレームワークとは，Shoup[48] によって形式化された，ハイブリッド暗号方式である．このフレームワークは，共通鍵暗号の利点である計算速度の速さを活かしつつ，欠点である鍵の長さや鍵交換の難しさを，KEM によってカバーする形になっている．具体的には，KEM 部が公開鍵暗号のような役割を果たし，公開鍵を用いて，暗号文および共通鍵を作成する．そして，KEM で出力された共通鍵を用いて，DEM 部が実際にメッセージの暗号化を行う仕組みとなっている．

2.5　鍵カプセル化メカニズム (PK-KEM)

鍵カプセル化メカニズム (PK-KEM)Π は以下の 3 つの確率的アルゴリズムからなる．

鍵生成 PKKEM.KG: セキュリティパラメータ 1^κ を入力とし，秘密鍵 sk と公開鍵 pk の対を出力する．

カプセル化 PKKEM.Encap: 公開鍵 pk を入力とし，暗号文 $c \in \mathcal{C}$，共通鍵 $k \in \mathcal{K}$ を出力する．

復号 PKKEM.Decap: 秘密鍵 sk，暗号文 c を入力とし，共通鍵 k (あるいは $\perp$) を出力する．

$\mathcal{K}$，$\mathcal{C}$ は Π の共通鍵空間，暗号文空間である．

正当性　PKKEM.KG から出力された全ての鍵ペア (sk, pk) に対し，以下を満たさなければならない．

$$\mathsf{PKKEM.Encap}(pk) = (c, k) \wedge \mathsf{PKKEM.Decap}(sk, c) = k$$

安全性の定義については [33] などでも詳細に述べられているが，そのうち提案手法で用いる以下の安全性について述べる．

2.5.1　OW-CCA 安全性

PK-KEM Π の適応的選択暗号文攻撃に対する一方向性 (One-wayness against Adaptive Chosen-Ciphertext Attacks, OW-CCA) は，以下の攻撃者 $\mathcal{A}$ と OW-CCA Challenger $\mathcal{CH}$ 間の IND-CCA game を用いて定義される．この定義は，例え攻撃対象以外の任意の暗号文に対して復号してもらうことが可能な環境にある攻撃者に対してであっても，暗号文から平文を完全に元に戻すことができない，という安全性の定義になっている．IND-CCA と異なり，1bit も漏れない，という意味ではないため，平文の部分的な情報が漏れることに関しては何も言うことができない定義となっている (IND より弱い安全性である)．

Setup. $\mathcal{CH}$ は $\mathsf{PKE.KG}(1^\kappa)$ を実行する．出力された pk を $\mathcal{A}$ に渡し，sk を保持しておく．

Phase 1. $\mathcal{A}$ は $\mathcal{CH}$ に対し，任意の回数，復号クエリ c を発行することができる．$\mathcal{CH}$ は c に対し，正しい復号結果 $k \leftarrow \mathsf{PKKEM.Decap}(sk, c)$ $(k \in \mathcal{K} \cup \perp)$ を返す．

Challenge. $\mathcal{CH}$ は $(c^*, k^*) \leftarrow \mathsf{PKKEM.Encap}(pk)$ を計算し，c^* を $\mathcal{A}$ に渡す．

Phase 2. $\mathcal{A}$ は Phase 1 と同様に復号クエリを発行することができる．ただし，Challenge の際受け取った暗号文 c^* を復号クエリとすることはできない．

Guess. $\mathcal{A}$ は c^* の復号結果の予測として k' を出力する．

ここで，ある PK-KEMΠ における $\mathcal{A}$ の OW-CCA アドバンテージを以下の様に定義する．

$$\mathsf{Adv}^{\text{OW-CCA}}_{\Pi,\mathcal{A}} = \Pr[k' = k^*]$$

定義 2.5. 全ての多項式時間アルゴリズム $\mathcal{A}$ に対し，$\mathsf{Adv}^{\text{OW-CCA}}_{\Pi,\mathcal{A}}$ が無視できるほど小さい時，Π は *OW-CCA* 安全であるという．

2.6　ID ベース鍵カプセル化メカニズム (IB-KEM)

ID ベース鍵カプセル化メカニズム (IB-KEM)Π は以下の 4 つの確率的アルゴリズムからなる．

セットアップ IBKEM.Setup: セキュリティパラメータ 1^κ を入力とし，マスター秘密鍵 msk と公開パラメータ prm の対を出力する．

鍵導出 IBKEM.Ext: 公開パラメータ prm，マスター秘密鍵 msk，$\mathsf{ID} \in \mathcal{I}$ を入力とし，ID に対する復号鍵 d_{ID} を出力する．

カプセル化 IBKEM.Encap: 公開パラメータ prm，$\mathsf{ID} \in \mathcal{I}$ を入力とし，暗号文 $c \in \mathcal{C}$，共通鍵 $k \in \mathcal{K}$ を出力する．

復号 IBKEM.Dec: 復号鍵 d_{ID}，暗号文 c を入力とし，共通鍵 k (あるいは $\perp$) を出力する．

$\mathcal{I}$，$\mathcal{K}$，$\mathcal{C}$ は Π の ID 空間，共通鍵空間，暗号文空間である．

正当性 IBKEM.Setup から出力された全ての $(\mathsf{msk}, \mathsf{prm})$,$\mathsf{ID} \in \mathcal{I}$，$d_{\mathsf{ID}} \leftarrow \mathsf{IBE.Ext}(\mathsf{prm}, \mathsf{msk}, \mathsf{ID})$ に対し，以下を満たさなければならない．

$$\mathsf{IBKEM.Encap}(\mathsf{ID}) = (c, k) \wedge \mathsf{IBKEM.Decap}(\mathsf{IBKEM.Ext}(\mathsf{prm}, \mathsf{msk}, \mathsf{ID}), c) = k$$

IB-KEM については，[6] などでも詳細に述べられているが，提案手法で用いる以下の 2 つの安全性の定義について詳細に述べる．

2.6.1　OW-ID-CPA 安全性

IB-KEM Π の選択平文攻撃および適応的選択 ID 攻撃に対する一方向性 (One-wayness against Adaptive ID and Adaptive Chosen-Plaintext Attacks, OW-ID-CPA) は，以下の攻撃者 $\mathcal{A}$ と OW-ID-CPA Challenger $\mathcal{CH}$ 間の OW-ID-CPA game を用いて定義される．この定義は，例え任意の平文に対する暗号文を得ることが可能な環境にある攻撃者に対してであっても，暗号文から平文を完全に元に戻すことができない，という安全性の定義になっている．やはり，IND-ID-CPA と異なり，1bit も漏れない，という意味ではないため，平文の部分的な情報が漏れることに関しては何も言うことができない定義となっている (IND より弱い安全性である)．

Setup. $\mathcal{CH}$ は $\mathsf{IBE.Setup}(1^\kappa)$ を実行する．出力された prm を $\mathcal{A}$ に渡し，msk を保持しておく．

Phase 1. $\mathcal{A}$ は $\mathcal{CH}$ に対し，任意の回数，鍵導出クエリ ID を発行することができる．$\mathcal{CH}$ はそれぞれのクエリ ID に対し，正しい復号鍵 $d_{\mathsf{ID}} \leftarrow \mathsf{IBE.Ext}(\mathsf{prm}, \mathsf{msk}, \mathsf{ID})$ を返す．

Challenge. $\mathcal{A}$ は鍵導出クエリとして発行していない ID^* を選び，$\mathcal{CH}$ に送る．$\mathcal{CH}$ は $(c^*, k^*) \leftarrow \mathsf{IBKEM.Encap}(\mathsf{prm}, \mathsf{ID}^*)$ を計算し，c^* を $\mathcal{A}$ に渡す．

Phase 2. $\mathcal{A}$ は Phase 1 と同様に鍵導出クエリを発行することができる．ただし，$\mathcal{A}$ は ID^* を鍵導出クエリとすることはできない．

Guess. $\mathcal{A}$ は c^* の復号結果の予測として k' を出力する．

ここで，ある IB-KEMΠ における $\mathcal{A}$ の OW-ID-CPA アドバンテージを以下の様に定義する．

$$\mathsf{Adv}^{\text{OW-ID-CPA}}_{\Pi,\mathcal{A}} = \Pr[k' = k^*]$$

定義 2.6. 全ての多項式時間アルゴリズム $\mathcal{A}$ に対し，$\mathsf{Adv}^{\text{OW-ID-CPA}}_{\Pi,\mathcal{A}}$ が無視できるほど小さいとき，Π は *OW-ID-CCA* 安全であるという．

2.6.2　IND-ID-CPA 安全性

IB-KEM Π の選択平文攻撃および適応的選択 ID 攻撃に対する識別不可能性 (IND-ID-CPA) は，以下の攻撃者 $\mathcal{A}$ と IND-ID-CPA Challenger $\mathcal{CH}$ 間の IND-ID-CPA game を用いて定義される．

Setup. $\mathcal{CH}$ は $\mathsf{IBKEM.Setup}(1^\kappa)$ を実行する．出力された prm を $\mathcal{A}$ に渡し，msk を保持しておく．

Phase 1. $\mathcal{A}$ は $\mathcal{CH}$ に対し，任意の回数，鍵導出クエリ ID を発行することができる．$\mathcal{CH}$ はそれぞれのクエリ ID に対し，正しい復号鍵 $d_{\mathsf{ID}} \leftarrow \mathsf{IBKEM.Ext}(\mathsf{prm}, \mathsf{msk}, \mathsf{ID})$ を返す．

Challenge. $\mathcal{A}$ は鍵導出クエリとして発行していない ID^* を選び，$\mathcal{CH}$ に送る．$\mathcal{CH}$ は $(c^*, k_0^*) \leftarrow \mathsf{IBKEM.Encap}(\mathsf{prm}, \mathsf{ID}^*)$ を計算する．また，$k_1^* \leftarrow \mathcal{K}$ をランダムに選ぶ．$\mathcal{CH}$ はランダムにコイン $b_C \in \{0, 1\}$ を振り，$(c^*, k_{b_c}^*)$ を $\mathcal{A}$ に渡す．

Phase 2. $\mathcal{A}$ は Phase 1 と同様に鍵導出クエリを発行することができる．ただし，$\mathcal{A}$ は ID^* を鍵導出クエリとすることはできない．

Guess. $\mathcal{A}$ は $\mathcal{CH}$ の選んだ b_C の予測として b_A を出力する．

ここで，ある IB-KEMΠ における $\mathcal{A}$ の IND-ID-CPA アドバンテージを以下の様に定義する．

$$\mathsf{Adv}^{\text{IND-ID-CPA}}_{\Pi,\mathcal{A}} = |\Pr[b_A = b_C] - 1/2|$$

定義 2.7. 全ての多項式時間アルゴリズム $\mathcal{A}$ に対し，$\mathsf{Adv}^{\text{IND-ID-CPA}}_{\Pi,\mathcal{A}}$ が無視できるほど小さいとき，Π は *IND-ID-CPA* 安全であるという．

2.6.3　乱数に関するターゲット衝突困難性

IB-KEM についても，提案手法の証明に必要となる乱数に関するターゲット衝突困難性を定義する．基本的には ID ベース暗号について定義したものと同義のものとなっている．

以下の攻撃者 $\mathcal{A}$ と Challenger $\mathcal{CH}$ 間のゲームで定義する．

$$\mathsf{Adv}^{\text{Rand}}_{\Pi,\mathcal{A}} = \Pr\left[\begin{array}{l}(\mathsf{msk}, \mathsf{prm}) \leftarrow \mathsf{IBKEM.Setup}(1^\kappa); R_{IBE} \leftarrow \mathcal{R}_{IBE};\\ (\mathsf{ID}', R'_{IBE}) \leftarrow \mathcal{A}(\mathsf{msk}, \mathsf{prm}, R_{IBE}) :\\ \mathsf{IBKEM.Encap}(\mathsf{prm}, \mathsf{ID}^*; R'_{IBE}) =\\ \mathsf{IBKEM.Encap}(\mathsf{prm}, \mathsf{ID}^*; R_{IBE}) \wedge R'_{IBE} \neq R_{IBE}\end{array}\right]$$

$\mathcal{R}_{IBE}$ は暗号化アルゴリズムの乱数空間を表す．

定義 2.8. 全ての多項式時間アルゴリズム $\mathcal{A}$ に対し，$\mathsf{Adv}^{\text{Rand}}_{\Pi,\mathcal{A}}$ が無視できるほど小さいとき，Π は乱数に関するターゲット衝突困難性を持つという．

2.7　タグベース鍵カプセル化メカニズム (TB-KEM)

タグベース鍵カプセル化メカニズム (TB-KEM) は PK-KEM の Encap, Decap の際に tag も入力するような形のものであり，[36] で詳細に述べられている．[4] で定義された ”Tag-KEM ”とは違う要素技術であることに注意されたい．

TB-KEM Π は以下の 3 つの確率的アルゴリズムからなる．

鍵生成 TBKEM.KG: セキュリティパラメータ 1^κ を入力とし，秘密鍵 sk と公開鍵 pk の対を出力する．

カプセル化 TBKEM.Encap: 公開鍵 pk，タグ tag を入力とし，暗号文 $c \in \mathcal{C}$，共通鍵 $k \in \mathcal{K}$ を出力する．

復号 TBKEM.Decap: 秘密鍵 sk，タグ tag，暗号文 c を入力とし，共通鍵 k (あるいは $\perp$) を出力する．

$\mathcal{K}$，$\mathcal{C}$ は Π の共通鍵空間，暗号文空間である．

正当性　TBKEM.KG から出力された全ての鍵ペア (sk, pk)，および全てのタグ tag に対し，以下を満たさなければならない．

$$\mathsf{TBKEM.Encap}(pk, tag) = (c, k) \wedge \mathsf{TBKEM.Decap}(sk, tag, c) = k$$

2.7.1　IND-stag-wCCA 安全性

TB-KEM Π に対する攻撃者 $\mathcal{A}$ の選択的タグ，弱選択暗号文攻撃に対する識別不可能性 (Indistinguishability against Selective tag and Adaptive weak Chosen-Plaintext Attacks, IND-stag-sCCA) は，以下の攻撃者 $\mathcal{A}$ と IND-stag-wCCA Challenger $\mathcal{CH}$ 間のゲームで定義する．この定義は，例え攻撃対象以外の任意の暗号文に対して復号してもらうことが可能な環境にある攻撃者に対してであっても，暗号文から平文の情報が 1bit も漏れない，という安全性の定義になっている．stag というのは，攻撃対象となる tag を最初に選び，後に変更することができない，という意味となっている．

Setup. $\mathcal{A}$ は攻撃対象となる tag^* を最初に選び，出力する．$\mathcal{CH}$ は $\mathsf{TBKEM.KG}(1^\kappa)$ を実行する．出力された pk を $\mathcal{A}$ に渡し，sk を保持しておく．

Phase 1. $\mathcal{A}$ は $\mathcal{CH}$ に対し，任意の回数，復号クエリ (c, tag) を発行することができる．$\mathcal{CH}$ はそれぞれのクエリ (c, tag) に対し，正しい復号結果 $k \leftarrow \mathsf{TBKEM.Decap}(sk, tag, c)$ を返す．ただし，$\mathcal{A}$ は $tag = tag^*$ となる tag を含むクエリを発行することはできない．

Challenge. $\mathcal{CH}$ は $(c^*, k_0^*) \leftarrow \mathsf{TBKEM.Encap}(pk, tag^*)$ を計算する．また，$k_1^* \leftarrow \mathcal{K}$ をランダムに選ぶ．$\mathcal{CH}$ はランダムにコイン $b_C \in \{0, 1\}$ を振り，$(c^*, k_{b_c}^*)$ を $\mathcal{A}$ に渡す．

Phase 2. $\mathcal{A}$ は Phase 1 と同様に復号クエリを発行することができる．ただし，$\mathcal{A}$ は $tag = tag^*$ となる tag を含む復号クエリを発行することはできない．

Guess. $\mathcal{A}$ は $\mathcal{CH}$ の選んだ b_C の予測として b_A を出力する．

定義 2.9. 全ての多項式時間アルゴリズム $\mathcal{A}$ に対し，$\mathsf{Adv}^{\text{IND-stag-wCCA}}_{\Pi,\mathcal{A}}$ が無視できるほど小さいとき，Π は *IND-stag-wCCA* 安全であるという．

2.8 Encapsulation 方式

Encapsulation 方式は Boneh と Katz [13] によって BK 変換の主たる構成要素として使うために導入された，ランダムな値をコミットするコミットメント方式の様なものである．この値は，後にデコミットメントの情報を用いて復元できる．

Encapsulation 方式 E は以下の 3 つのアルゴリズムからなる．

セットアップ ESetup: セキュリティパラメータ 1^κ を入力とし，公開パラメータ $\mathsf{prm_E}$ の対を出力する．

コミットメント Com: 公開パラメータ $\mathsf{prm_E}$ を入力とし，コミットされた値 $r \in \mathcal{V}$，コミットメント $com \in \mathcal{COM}$，デコミットメント $d \in \mathcal{D}$ を出力する．

リカバー Rec: 公開パラメータ $\mathsf{prm_E}$，コミットメント com，デコミットメント d を入力とし，コミットされた値 r (あるいは $\perp$) を出力する．

$\mathcal{V}$，$\mathcal{COM}$，$\mathcal{D}$ はそれぞれ E のコミットされた値の空間，コミットメントの空間，デコミットメントの空間である．

ESetup から出力された全ての $\mathsf{prm_E}$，及び $\mathsf{Com}(\mathsf{prm_E})$ から出力された全ての $(r, com, d) \in \mathcal{V} \times \mathcal{COM} \times \mathcal{D}$ について，$\mathsf{Rec}(\mathsf{prm_E}, com, d) = r$ が成り立たねばならない．

Encapsulation 方式には以下の Hiding と Binding と呼ばれる安全性が要求される．

2.8.1 Hiding

Encapsulation 方式 E の Hiding 性に対する攻撃者 $\mathcal{A}$ のアドバンテージを以下の様に定義する．直感的には，デコミットメント d なしにコミットメント com からコミットされた値 r の情報が漏れないという安全性を表している．

$$\mathsf{Adv}^{\text{Hiding}}_{E,\mathcal{A}} = \left| \Pr \left[\begin{array}{ll} b_C \leftarrow \{0,1\};\ \mathsf{prm_E} \leftarrow \mathsf{ESetup}(1^\kappa); & \\ (r_1^*, com^*, d^*) \leftarrow \mathsf{Com}(\mathsf{prm_E}); & : b_A = b_C \\ r_0^* \leftarrow \mathcal{V};\ b_A \leftarrow \mathcal{A}(\mathsf{prm_E}, r^*_{b_C}, com^*) & \end{array} \right] - \frac{1}{2} \right|$$

定義 2.10. 全ての多項式時間アルゴリズム $\mathcal{A}$ に対し，$\mathsf{Adv}^{\text{Hiding}}_{E,\mathcal{A}}$ が無視できるほど小さいとき，E は *hiding* であるという．

2.8.2 Binding

Encapsulation 方式 E の Binding 性に対する攻撃者 $\mathcal{A}$ のアドバンテージを以下の様に定義する．直感的には，あるコミットメント com リをカバーした結果が，$\perp$ でない異なる結果となるようなデコミットメント d を，いかなる攻撃者も作ることができないという安全性を表している．

$$\mathsf{Adv}^{\mathrm{Binding}}_{E,\mathcal{A}} = \Pr\left[\begin{array}{l}\mathsf{prm}_{\mathsf{E}} \leftarrow \mathsf{ESetup}(1^{\kappa}); (r^*, com^*, d^*) \leftarrow \mathsf{Com}(\mathsf{prm}_{\mathsf{E}});\\ d' \leftarrow \mathcal{A}(\mathsf{prm}_{\mathsf{E}}, com^*, d^*) :\\ \mathsf{Rec}(\mathsf{prm}_{\mathsf{E}}, com^*, d') \notin \{\perp, r^*\} \wedge d' \neq d^*\end{array}\right]$$

定義 2.11. 全ての多項式時間アルゴリズム $\mathcal{A}$ に対し $\mathsf{Adv}^{\mathrm{Binding}}_{E,\mathcal{A}}$ が無視できるほど小さいとき，E は *binding* であるという．

2.9 共通鍵暗号

共通鍵暗号 (Symmetric Key Encryption, SKE) 方式 Π は以下の 2 つのアルゴリズムからなる．

暗号化 SKE.Enc: 共通鍵 k 及びメッセージ m を入力として受け取り，k の下で m に対する暗号文 c を出力する．

復号 SKE.Dec: 共通鍵 k，暗号文 c を入力として受け取り，正しい復号結果 m(または $\perp$) を出力する．

正当性　全ての鍵 k に対し，以下を満たさなければならない．

$$\mathsf{SKE.Dec} = (k, \mathsf{SKE.Enc}(k, m)) = m$$

2.9.1 IND-COA 安全性

共通鍵暗号 Π に対する攻撃者 $\mathcal{A}$ の暗号文単独攻撃に対する識別不可能性 (Indistinguishability against Ciphertext Only Attacks, IND-COA) は，以下の攻撃者 $\mathcal{A}$ と IND-COA Challenger $\mathcal{CH}$ 間のゲームで定義する．この定義は，暗号文を見ても平文を完全に元に戻すことはできない，という定義である．

Challenge. $\mathcal{A}$ は 2 つの任意の平文 $m_0,\ m_1$ を選び，$\mathcal{CH}$ に送る．$\mathcal{CH}$ は $k^* \leftarrow \mathcal{K}$ をランダムに選ぶ．その後，$\mathcal{CH}$ はランダムにコイン $b_C \in \{0, 1\}$ を振り，$c^* \leftarrow \mathsf{SKE.Enc}(k^*, m_{b_C})$ を $\mathcal{A}$ に渡す．

Guess. $\mathcal{A}$ は $\mathcal{CH}$ の選んだ b_C の予測として b_A を出力する．

ここで，ある公開鍵暗号 Π における $\mathcal{A}$ の IND-CCA アドバンテージを以下の様に定義する．

$$\mathsf{Adv}^{\text{IND-COA}}_{\Pi,\mathcal{A}} = |\Pr[b_A = b_C] - 1/2|$$

定義 2.12. 全ての多項式時間アルゴリズム $\mathcal{A}$ に対し，$\mathsf{Adv}^{\text{IND-COA}}_{\Pi,\mathcal{A}}$ が無視できるほど小さい時，Π は *IND-COA* 安全であるという．

2.10　メッセージ認証子 (MAC)

メッセージ認証子 (Message Authentication Code, MAC) 方式 Σ は以下の 2 つのアルゴリズムからなる．

MAC タグ生成 MAC**:** MAC 鍵 mk 及びメッセージ m を入力として受け取り，mk の下で m に対する正当な MAC タグ σ を出力する．

検証 $\mathsf{Verify}_{\mathsf{MAC}}$**:** MAC 鍵 mk，メッセージ m，MAC タグ σ を入力として受け取り，正しい MAC タグならば `accept` を，そうでなければ `reject` を出力する．

正当性　全ての MAC 鍵 mk，および全ての $m \in \mathcal{M}$ に対し，以下を満たさなければならない．

$$\mathsf{Verify}_{\mathsf{MAC}}(mk, \mathsf{Sign}(mk, m)) = \mathtt{accept}$$

2.10.1　SUF-OT$_{MAC}$

MACΣ に対する攻撃者 $\mathcal{A}$ の一回のみの強偽造不可能性 (SUF-OT$_{MAC}$) は，以下の攻撃者 $\mathcal{A}$ と SUF-OT$_{MAC}$ Challenger $\mathcal{CH}$ 間のゲームで定義する．この定義は，例え高々 1 回のみ，平文に対する MAC タグのクエリを発行することができる攻撃者に対してであっても，発行した (m, σ) ではなく，かつ $\mathsf{Verify}_{\mathsf{MAC}}$ が `accept` となるような平文と署名の組を偽造することはできない，という安全性の定義となっている．

Setup. $\mathcal{CH}$ は MAC 鍵 mk をランダムに選ぶ．

Query. $\mathcal{A}$ は $\mathcal{CH}$ に対し，高々1 回のみの MAC タグ生成クエリ m を発行することができる．$\mathcal{CH}$ は $\mathcal{A}$ に対し，正しい MAC タグ $\sigma \leftarrow \mathsf{MAC}(mk, m)$ を $\mathcal{A}$ に渡す．

Forge. $\mathcal{A}$ は平文と MAC タグの組 (m^*, σ^*) を出力する

ここで，ある MACΠ のアドバンテージを以下のように定義する．

$$\mathsf{Adv}^{\text{SUF-OT}_{\text{MAC}}}_{\Sigma,\mathcal{A}} = \Pr[\mathsf{Verify}_{\mathsf{MAC}}(mk, m^*, \sigma^*) = \mathit{accept} \wedge (m, \sigma) \neq (m^*, \sigma^*)]$$

定義 2.13. 全ての多項式時間アルゴリズム $\mathcal{A}$ に対し，$\mathsf{Adv}^{\text{SUF-OT}_{\text{MAC}}}_{\Sigma,\mathcal{A}}$ が無視できるほど小さい時，Π は $SUF\text{-}OT_{MAC}$ 安全という.

Chapter 3　時限式暗号/時間前開封機能付き時限式暗号

本章では，本研究の提案手法である，時間前開封機能付き時限式暗号 (TRE-PC)，およびその元となっている時限式暗号 (TRE) について，それぞれのアルゴリズムや安全性の定義について振り返る．

3.1　時限式暗号 (TRE)

時限式暗号 (Timed-Release Encryption, TRE) とは，1993 年に May[40] によって提案された機能付き暗号の 1 つであり，正しい受信者であっても，送信者が指定した時刻になるまでは復号することができない暗号のことである．

3.1.1　TRE の構成へのアプローチ

TRE の機能を満たす暗号を構築するアイデアとして，以下の二つのものがある．

- Time-lock puzzles
- Trusted decryption agents

Time-lock puzzles は，受信者のコンピュータで復号するためには，暗号を受け取った時点から復号を始めたとしても，少なくとも送信者の指定した時間までには計算を終えることができず，復号することができない，という暗号を作成することで TRE の機能を達成するという手法である ([45][38] など)．この手法は，受信者に対して高い計算コストを強いる上に，復号にかかる時間を正しく予測することが困難であるため，定められた時間に確かに復号できるようになると保証することが難しい．

一方，Trusted decryption agents を用いた手法は，受信者に対して秘密をもった暗号を送信者が作成した上で，信頼できる第三者機関にその秘密情報を渡しておく．そして，指定された時間になった時点で，第三者機関が受信者に対してその秘密情報を発行することで，はじめてその暗号を復号できるようになるという手法である ([27][20][22][24] など)．この手法では，受信者は高い計算コストを強いられることもなく，確実に正しい時刻に復号をすることが可能となる．また，送信者は受診者者が復号にかかる時間を予測するということもしなくてよい．ただし，第三者機関が信頼できなければならないし，指定された時間に利用できる状態である必要があるという問題が新たに生じる．

本稿では多くの TRE で考えられている，第三者機関を用いた TRE の手法について紹介を行う．

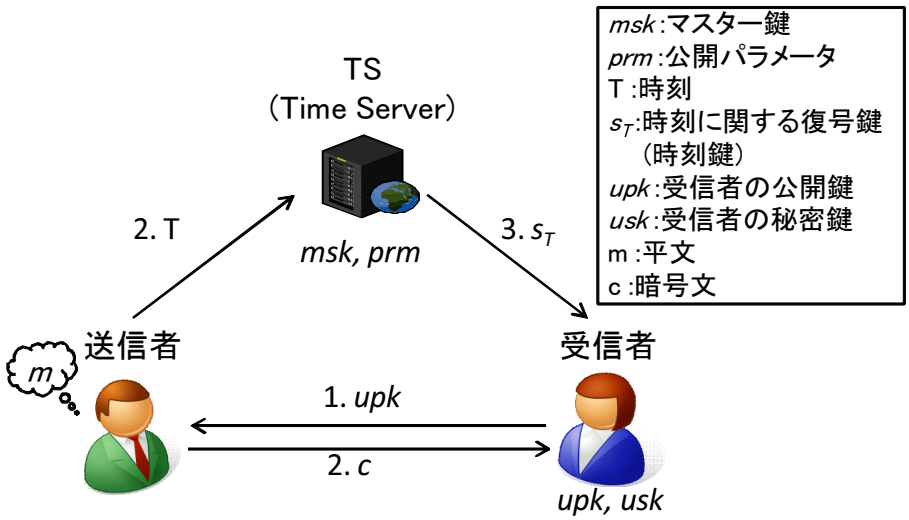

図 3.1: TRE のモデル図

3.1.2　TRE のアルゴリズム

Time Server(以下 TS とする) を用いた TRE についてのモデル図を図 3.1 に示す．TRE 方式 Π は，以下に述べる 5 つのアルゴリズムから成る．

TRE.Setup: セキュリティパラメータ 1^κ を入力とし，TS のマスター鍵 msk と公開パラメータ prm の対を出力する．

TRE.Ext：公開パラメータ prm，マスター鍵 msk，時間 T を入力とし，時刻 T に関する秘密鍵 (以下，時刻鍵)s_T を出力する．

TRE.UKG：セキュリティパラメータ 1^κ を入力とし，ユーザーの秘密鍵，及び公開鍵の組 (usk, upk) を出力する．

TRE.Enc: 公開パラメータ prm，公開鍵 upk，復号を行わせたい時刻 T，平文 m を入力とし，暗号文 c を出力する．

TRE.Dec: 公開パラメータ prm，秘密鍵 usk，時刻鍵 s_T，暗号文 c を入力とし，平文 (あるいは復号失敗 “$\perp$”) $m \cup \{\perp\}$ を出力する．

ただし，全ての $(\mathsf{msk}, \mathsf{prm}) \leftarrow \mathsf{TRE.Setup}$, $(sk, pk) \leftarrow \mathsf{TRE.UKG}$, $s_T \leftarrow \mathsf{TRE.Ext}(\mathsf{prm}, \mathsf{msk}, T)$，および全ての $m \in \mathcal{M}$ について，$\mathsf{TRE.Dec}(\mathsf{prm}, usk, s_T, \mathsf{TRE.Enc}(\mathsf{prm}, upk, T, m)) = m$ が成り立つことが要求される．また，$\mathcal{M}$，$\mathcal{C}$，$\mathcal{T}$ はそれぞれ，Π の平文空間，暗号文空間，時間空間である．

3.1.3　TRE における安全性の定義

TRE において考えなければならない安全性は以下の 3 つの安全性である．

- 外部者に対する安全性 (Outsider Security)
- TS に対する安全性 (Time Server Security)
- 受信者に対する安全性 (Insider Security)

ただし，外部者に対する安全性は，TS に対する安全性に内包されることがわかっているため，本稿ではその他の 2 つについて詳細に述べる．

なお，本稿では [25] で述べられている安全性の定義を基本として TRE に関する安全性の定義を述べることにする．

TS に対する安全性

たとえマスター鍵 msk を持っている TS であっても，秘密鍵 usk なしに暗号文から平文の情報を得られてはならない．そのために必要となる安全性の定義を以下のゲームで表す．

Setup. $\mathcal{CH}$ は $\mathsf{TRE.UKG}(1^\kappa)$ を実行し，(usk, upk) を，また，$\mathsf{TRE.Setup}$ を実行し，$(\mathsf{msk}, \mathsf{prm})$ を得る．出力された $\mathsf{msk}, \mathsf{prm}, upk$ を $\mathcal{A}$ に渡し，usk を保持しておく．

Phase 1. $\mathcal{A}$ は $\mathcal{CH}$ に対し，任意の回数，暗号文と時刻の組 (c, T) を復号クエリとして発行することができる．$\mathcal{CH}$ はそれぞれの復号クエリに対して，まず $s_T \leftarrow \mathsf{TRE.Ext}(\mathsf{prm}, \mathsf{msk}, T)$ を実行し，得られた時刻鍵 s_T を用いて $m \leftarrow \mathsf{TRE.Dec}(\mathsf{prm}, usk, s_T, c)$ を $\mathcal{A}$ に返す．

Challenge. $\mathcal{A}$ は 2 つの任意の平文 m_0, m_1, および時刻 T^* を選び，$\mathcal{CH}$ に送る．$\mathcal{C}$ はランダムにコイン $b_C \in \{0, 1\}$ を振り，m_{b_C} の暗号文 $c^* \leftarrow \mathsf{Enc}(\mathsf{prm}, upk, T^*, m_{b_C})$ を計算し，c^* を $\mathcal{A}$ に渡す．

Phase 2. $\mathcal{A}$ は Phase 1 と同様に，任意の回数復号クエリを発行することができる．ただし，(c^*, T^*) の組を復号クエリとして発行することはできない．

Guess. $\mathcal{A}$ は $\mathcal{CH}$ の選んだ b_C として b_A を出力する．

ここで，ある TREΠ 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．

$$\mathsf{Adv}^{\text{IND-TRE-CCA}_{\text{TS}}}_{\Pi,\mathcal{A}} = |\Pr[b_A = b_C] - 1/2|$$

定義 3.1. 全ての多項式時間アルゴリズム $\mathcal{A}$ に対し，$\mathsf{Adv}^{\text{IND-TRE-CCA}_{\text{TS}}}_{\Pi,\mathcal{A}}$ が無視できるほど小さい時，Π は *IND-TRE-CCA*$_{TS}$ 安全であるという．

なお，ここで $\mathcal{A}$ が $\mathcal{CH}$ に T をクエリとして聞かないのは，$\mathcal{A}$ は msk を持っているため，自分で任意の T に対して TRE.Ext を実行し，時刻鍵 s_T を得ることが可能だからである．

この安全性が達成されている時，マスター鍵 msk を持っている TS に対しても，平文に関する情報が漏れない，ということを保証することができる．

受信者に対する安全性

たとえマスター鍵 msk を持っている TS であっても，秘密鍵 usk なしに暗号文を復号できてはならない．

TRE がその機能を正しく果たすためには，秘密鍵 usk を持つ正しい受信者であっても，時刻鍵なしに指定された時刻以前に暗号文から平文の情報を得られてはならない．そのために必要となる安全性の定義を以下のゲームで表す．

Setup. $\mathcal{CH}$ は $\mathsf{TRE.Setup}(1^\kappa)$，$\mathsf{TRE.UKG}(1^\kappa)$ を実行し，$(\mathsf{msk}, \mathsf{prm})$ および (usk, upk) を得る．出力された prm, usk, upk を $\mathcal{A}$ に渡し，msk を保持しておく．

Phase 1. $\mathcal{A}$ は $\mathcal{CH}$ に対し，時刻 T を時刻鍵導出クエリとして $\mathcal{CH}$ に発行することができる．$\mathcal{CH}$ はそれぞれの時刻鍵導出クエリに対して，$s_T \leftarrow \mathsf{TRE.Ext}(\mathsf{prm}, \mathsf{msk}, T)$ を実行し，その時刻に対する正しい時刻鍵 s_T を $\mathcal{A}$ に返す．

Challenge. $\mathcal{A}$ は 2 つの任意の平文 m_0, m_1, および時刻 T^* を選び，$\mathcal{CH}$ に送る．$\mathcal{C}$ はランダムにコイン $b_C \in \{0, 1\}$ を振り，m_{b_C} の暗号文 $c^* \leftarrow \mathsf{TRE.Enc}(upk, T^*, m_{b_C})$ を計算し，c^* を $\mathcal{A}$ に渡す．

Phase 2. $\mathcal{A}$ は Phase 1 と同様に，任意の回数時刻鍵導出クエリを発行することができる．ただし，$T \geq T^*$ となるような T をクエリとして発行することはできない．

Guess. $\mathcal{A}$ は $\mathcal{CH}$ の選んだ b_C として b_A を出力する．

ここで，ある TREΠ 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．

$$\mathsf{Adv}^{\text{IND-TRE-CPA}_{\text{IS}}}_{\Pi,\mathcal{A}} = |\Pr[b_A = b_C] - 1/2|$$

定義 3.2. 全ての多項式時間アルゴリズム $\mathcal{A}$ に対し，$\mathsf{Adv}^{\text{IND-TRE-CCA}_{\text{TS}}}_{\Pi,\mathcal{A}}$ が無視できるほど小さい時，Π は *IND-TRE-CCA*$_{IS}$ 安全であるという．

なお，この安全性において'CCA' でなく'CPA' について考えている理由は，$\mathcal{A}$ は usk を持っているため，時刻鍵 s_T さえ得ることができればすべての暗号文について自身で復号することができるため，復号クエリを発行する必要がないと考えられるからである．

この安全性が達成されている時，秘密鍵 usk を持っている受信者であっても，時刻鍵 s_T なしに暗号文から平文に関する情報が漏れない，ということを保証することができる．

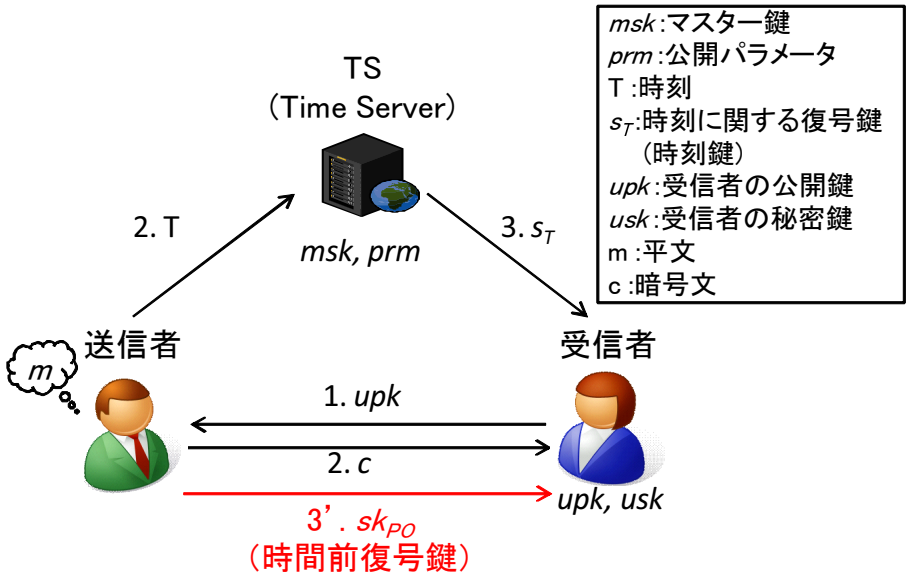

図 3.2: TRE-PC のモデル図

3.2　時間前開封機能付き時限式暗号 (TRE-PC)

時間前開封機能付き時限式暗号 (Timed-Release Encryption with Pre-open Capability, TRE-PC) は，2005 年に Hwang ら [35] によって提案された時限式暗号の一種である．

一般的な TRE では，送信者が暗号文を送った後に，何らかの理由で指定した時刻以前に情報を開示したいことがあったとしても，受信者は最初に指定された時刻が来るまでは復号を行うことはできないため，送信者は開封時刻を変更してもう一度暗号文を送信する必要がある．

そこで TRE-PC では，送信者によって定められた時刻に TS によって送られる秘密情報だけでなく，送信者によって送られる時間前復号鍵 (Pre-open Key) と呼ばれる秘密情報によっても暗号文を復号することができる．この機能があることにより，TRE をより柔軟に用いることができると考えられる．当然，正しい受信者以外が時間前復号鍵を受け取ったとしても，暗号文から平文の情報を一切得ることができてはならないし，正しい受信者であっても，指定時刻になる，もしくは時間前復号鍵を受け取るまでは，暗号文を復号することができてはならない．

TRE-PC は Hwang らによって提案された後，Dent ら [28] によって 2007 年に定義が見直されている．本稿では，[28] の定義を用いている．

3.2.1　TRE-PC のアルゴリズム

Time Server(以下 TS とする) を用いた TRE についてのモデル図を図 3.2 に示す．TRE-PC 方式 Π は以下の 6 つのアルゴリズムからなる．

TRE.Setup: セキュリティパラメータ 1^κ を入力とし，TS のマスター鍵 msk と公開パラ

メータ prm の対を出力する．

TRE.Ext：公開パラメータ prm，マスター鍵 msk，時刻 $T \in \mathcal{T}$ を入力とし，時刻鍵 s_T を出力する．

TRE.UKG：セキュリティパラメータ 1^κ を入力とし，ユーザーの秘密鍵と公開鍵 (usk, upk) を出力する．

TRE.Enc: 公開パラメータ prm，公開鍵 upk，復号を行わせたい時刻 T，平文 $m \in \mathcal{M}$ を入力とし，暗号文 $c \in \mathcal{C}$，および時間前開封鍵 sk_{PO} を出力する．

$\mathsf{TRE.Dec_{TR}}$: 公開パラメータ prm，秘密鍵 usk，時刻鍵 s_T，暗号文 c を入力とし，平文 (あるいは復号失敗 “$\perp$”) $m \cup \{\perp\}$ を出力する．

$\mathsf{TRE.Dec_{PO}}$: 公開パラメータ prm，秘密鍵 usk，時間前開封鍵 sk_{PO}，暗号文 c を入力とし，平文 (あるいは復号失敗 “$\perp$”) $m \cup \{\perp\}$ を出力する．

ただし，全ての $(\mathsf{msk}, \mathsf{prm}) \leftarrow \mathsf{TRE.Setup}$，$(sk, pk) \leftarrow \mathsf{TRE.UKG}$，$sk_{PO}, s_T \leftarrow \mathsf{TRE.Ext}(\mathsf{prm}, \mathsf{msk}, T)$，および全ての $m \in \mathcal{M}$ について，$\mathsf{TRE.Dec_{TR}}(\mathsf{prm}, usk, s_T, \mathsf{TRE.Enc}(\mathsf{prm}, upk, T, m)) = m$，および，$\mathsf{TRE.Dec_{PO}}(\mathsf{prm}, usk, sk_{PO}, \mathsf{TRE.Enc}(\mathsf{prm}, upk, T, m)) = m$ が成り立つことが要求される．また，$\mathcal{M}$，$\mathcal{C}$，$\mathcal{T}$ はそれぞれ，Π の平文空間，暗号文空間，時間空間である．

3.2.2　TRE-PC に求められる安全性の定義

TRE-PC においても，TRE と同様に安全性の定義が考えられている．本稿では Dent らによって考えられた以下の 4 つの安全性の定義を用いている [28].

- 外部者に対する安全性 (Outsider Security)
- TS に対する安全性 (Time Server Security)
- 受信者に対する安全性 (Insider Security)
- 送信者に対する安全性 (Binding)

ただし，[28] の中で，外部者に対する安全性は TS に対する安全性に内包されていることが証明されているため，本節ではそれを除いた 3 つについての安全性の定義を述べる．

TS に対する安全性

マスター鍵を持った TS であっても，秘密鍵がなければ暗号文から平文の情報を得ることができないという安全性．以下のゲームで定義される．

Setup. $\mathcal{CH}$ は $\mathsf{TRE.Setup}(1^\kappa)$，および $\mathsf{TRE.UKG}(1^\kappa)$ を実行する．出力された $\mathsf{prm}, \mathsf{msk}, upk$ を $\mathcal{A}$ に渡し，usk を保持しておく．

Phase 1. $\mathcal{A}$ は $\mathcal{CH}$ に対し，任意の回数，暗号文と時間前開封鍵の組 (c, sk_{PO})，および，暗号文と時刻の組 (c, T) を復号クエリとして発行することができる．前者を時間前復号クエリ，後者を指定時間復号クエリと呼ぶこととする．$\mathcal{CH}$ はそれぞれのクエリに対し，前者は $m \leftarrow \mathsf{TRE.Dec_{PO}}(\mathsf{prm}, usk, sk_{PO}, c)$ を，後者はまず，$s_T \leftarrow \mathsf{TRE.Ext}(\mathsf{prm}, \mathsf{msk}, T)$ を実行し，それを用いて $m \leftarrow \mathsf{TRE.Dec_{TR}}(\mathsf{prm}, usk, s_T, c)$ を返す

Challenge. $\mathcal{A}$ は 2 つの任意の平文 m_0，m_1 および時刻 T^* を選び，$\mathcal{CH}$ に送る．$\mathcal{CH}$ はランダムにコイン $b_C \leftarrow \{0,1\}$ を振り，m_{b_C} の暗号文 $(c^*, sk^*_{PO}) \leftarrow \mathsf{TRE.Enc}(\mathsf{prm}, upk, T^*, m_{b_C})$ を計算し，(c^*, sk^*_{PO}) を $\mathcal{A}$ に渡す．

Phase 2. $\mathcal{A}$ は Phase 1 と同様に，任意の回数の復号クエリを発行することができる．ただし，$(c, sk_{PO}) = (c^*, sk^*_{PO})$，$(c, T) = (c^*, T^*)$ となるクエリを発行することはできない．

Guess. $\mathcal{A}$ は $\mathcal{CH}$ の選んだ b_C の予測として b_A を出力する．$b_A = b_C$ のとき $\mathcal{A}$ はゲームに勝利したという．

ここで，$\mathcal{A}$ が得るアドバンテージを以下のように定義する．

$$\mathsf{Adv}^{\text{IND-TRPC-CCA}_{\text{TS}}}_{\Pi,\mathcal{A}} = |\Pr[b_A = b_C] - 1/2|$$

定義 3.3. 全ての多項式時間アルゴリズム $\mathcal{A}$ に対し，$\mathsf{Adv}^{\text{IND-TRPC-CCA}_{\text{TS}}}_{\Pi,\mathcal{A}}$ が無視できるほど小さい値であるとき，Π は *IND-TRPC-CCA*$_{TS}$ 安全であるという．

この安全性を考える上で，TRE と異なる点は，指定時間復号クエリだけでなく，時間前復号クエリも発行することができるという点である．また，注意しておきたいのは，Challenge において，攻撃者に暗号文だけではなく，時間前開封鍵も与えるという点である．

受信者に対する安全性

ユーザーの秘密鍵を持った受信者が，時間前復号鍵，もしくは時刻鍵を持たなければ，時間前に暗号文から平文の情報を得ることができないという安全性．以下のゲームで定義される．

Setup. $\mathcal{CH}$ は $\mathsf{TRE.Setup}(1^\kappa)$，および $\mathsf{TRE.UKG}(1^\kappa)$ を実行する．出力された prm, usk, upk を $\mathcal{A}$ に渡し，msk を保持しておく．

Phase 1. $\mathcal{A}$ は $\mathcal{CH}$ に対し，任意の回数，時刻 T を時刻鍵導出クエリとして $\mathcal{CH}$ に発行することができ，$\mathcal{CH}$ はそのクエリに対し，正しい時刻鍵 $s_T \leftarrow \mathsf{TRE.Ext}(\mathsf{prm}, \mathsf{msk}, T)$ を返す．

Challenge. $\mathcal{A}$ は 2 つの任意の平文 $m_0,\ m_1$, および時刻 T^* を選び，$\mathcal{CH}$ に送る．$\mathcal{CH}$ はランダムにコイン $b_C \in \{0,1\}$ を振り，m_{b_c} の暗号文 $(c^*, sk^*_{PO}) \leftarrow \mathsf{TRE.Enc}(\mathsf{prm}, upk, T^*, m_{b_c})$ を計算し，c^* を $\mathcal{A}$ に渡す．ここで選ぶ T^* は，Phase 1 で聞いたどの T よりも大きくなければならない．

Phase 2. $\mathcal{A}$ は Phase 1 と同様に，任意の回数の時間鍵導出クエリを発行することができる．ただし，$T \geq T^*$ となる T をクエリとして発行することはできない．

Guess. $\mathcal{A}$ は $\mathcal{CH}$ の選んだ b_C の予測として b_A を出力する．$b_A = b_C$ のとき $\mathcal{A}$ はゲームに勝利したという．

ここで，$\mathcal{A}$ が得るアドバンテージを以下のように定義する．

$$\mathsf{Adv}^{\text{IND-TRPC-CPA}_{\text{IS}}}_{\Pi,\mathcal{A}} = |\Pr[b_A = b_C] - 1/2|$$

定義 3.4. 全ての多項式時間アルゴリズム $\mathcal{A}$ に対し，$\mathsf{Adv}^{\text{IND-TRPC-CPA}_{\text{IS}}}_{\Pi,\mathcal{A}}$ が無視できるほど小さい値であるとき，Π は *IND-TRPC-CPA*$_{IS}$ 安全であるという．

Binding

送信者は，時刻 T に関して暗号化した暗号文 c について，時刻鍵 s_T を用いて $\mathsf{TRE.Dec_{PO}}$ で復号した結果と，時間前開封鍵 sk_{PO} を用いて $\mathsf{TRE.Dec_{TR}}$ で復号した結果が異なる平文が出力されるような暗号文を作成することができないという安全性．以下のゲームで定義される．

Setup. $\mathcal{CH}$ は $\mathsf{TRE.Setup}(1^\kappa)$，および $\mathsf{TRE.UKG}(1^\kappa)$ を実行する．出力された prm, upk を $\mathcal{A}$ に渡し，msk, usk を保持しておく．

Query. $\mathcal{A}$ は $\mathcal{CH}$ に対し，任意の回数，時間前復号クエリ (c, sk_{PO})，および指定時間復号クエリ (c, T) を発行することができ，$\mathcal{CH}$ はそれぞれのクエリに対し，TS に対する安全性のゲームと同様に正しい復号結果 $m \leftarrow \mathsf{TRE.Dec_{PO}}(\mathsf{prm}, usk, sk_{PO}, c), m \leftarrow \mathsf{TRE.Dec_{TR}}(\mathsf{prm}, usk, s_T, c)$ を返す．同様に，$\mathcal{A}$ は $\mathcal{CH}$ に対し，任意の回数，時刻 T を時刻鍵導出クエリとして $\mathcal{CH}$ に発行することができ，$\mathcal{CH}$ はそのクエリに対し，正しい時刻鍵 $s_T \leftarrow \mathsf{TRE.Ext}(\mathsf{prm}, \mathsf{msk}, T)$ を返す．

Output. $\mathcal{A}$ は (c^*, T^*, sk^*_{PO}) の組を出力する．

ここで，$\mathcal{A}$が得るアドバンテージを以下のように定義する．

$$\mathsf{Adv}^{\mathrm{BINDING}}_{\Pi,\mathcal{A}} = \Pr[\perp \neq \mathsf{Dec}_{\mathsf{PO}}(\mathsf{prm}, usk, sk^*_{PO}, c^*) \neq \mathsf{Dec}_{\mathsf{TR}}(\mathsf{prm}, usk, s^*_T, c^*) \neq \perp]$$

定義 3.5. 全ての多項式時間アルゴリズム $\mathcal{A}$ に対し，$\mathsf{Adv}^{\mathrm{BINDING}}_{\Pi,\mathcal{A}}$ が無視できるほど小さい値であるとき，Π は *Binding* の安全性を満たすという．

3.3 TRE-PC KEM

[28] において，TRE-PC KEM も考案されている．本節では TRE-PC KEM のアルゴリズムおよび安全性の定義について述べる．

3.3.1 TRE-PC KEM のアルゴリズム

TRE-PC KEMΠ は，以下の 6 つのアルゴリズムからなる．TRE-PC では $\mathsf{Enc}, \mathsf{Dec}$ だった部分が，TRE-PC KEM では $\mathsf{Encap}, \mathsf{Decap}$ に変わっている．

TRKEM.Setup: セキュリティパラメータ 1^κ を入力とし，TS のマスター鍵 msk と公開パラメータ prm の対を出力する．

TRKEM.Ext：公開パラメータ prm，マスター鍵 msk，時刻 $T \in \mathcal{T}$ を入力とし，時刻鍵 s_T を出力する．

TRKEM.UKG：セキュリティパラメータ 1^κ を入力とし，ユーザーの秘密鍵と公開鍵 (usk, upk) を出力する．

TRKEM.Encap: 公開パラメータ prm，公開鍵 upk，復号を行わせたい時刻 T を入力とし，暗号文 $c \in \mathcal{C}$，共通鍵 $k \in \mathcal{K}$，および時間前開封鍵 sk_{PO} を出力する．

$\mathsf{TRKEM.Decap}_{\mathsf{TR}}$: 公開パラメータ prm，秘密鍵 usk，時刻鍵 s_T，暗号文 c を入力とし，共通鍵 k(あるいは復号失敗 “$\perp$”) を出力する．

$\mathsf{TRKEM.Decap}_{\mathsf{PO}}$: 公開パラメータ prm，秘密鍵 usk，時間前開封鍵 sk_{PO}，暗号文 c を入力とし，共通鍵 k(あるいは復号失敗 “$\perp$”) を出力する．

ただし，全ての $(\mathsf{msk}, \mathsf{prm}) \leftarrow \mathsf{TRKEM.Setup}$, $(sk, pk) \leftarrow \mathsf{TRKEM.UKG}, sk_{PO}$, および $s_T \leftarrow \mathsf{TRKEM.Ext}(\mathsf{prm}, \mathsf{msk}, T)$ について，$\mathsf{TRKEM.Encap}(\mathsf{prm}, upk, T) = (c, k)$, $\mathsf{TRKEM.Decap}_{\mathsf{TR}}(\mathsf{prm}, usk, s_T, c) = k$，および，$\mathsf{TRKEM.Decap}_{\mathsf{PO}}(\mathsf{prm}, usk, sk_{PO}, c) = k$ が成り立つことが要求される．また，$\mathcal{M}$，$\mathcal{C}$，$\mathcal{T}$ はそれぞれ，Π の平文空間，暗号文空間，時間空間である．

3.3.2　TRE-PC KEM の安全性

TRE-PC KEM の安全性は根幹については TRE-PC と同様である．TRE-PC KEM においても以下の 3 つの定義が考えられている [28].

TS に対する安全性

マスター鍵を持った TS であっても，秘密鍵がなければ暗号文から平文の情報を得ることができないという安全性．以下のゲームで定義される．

Setup. $\mathcal{CH}$ は $\mathsf{TRKEM.Setup}(1^\kappa)$，および $\mathsf{TRKEM.UKG}(1^\kappa)$ を実行する．出力された $\mathsf{prm}, \mathsf{msk}, upk$ を $\mathcal{A}$ に渡し，usk を保持しておく．

Phase 1. $\mathcal{A}$ は $\mathcal{CH}$ に対し，任意の回数，暗号文と時間前開封鍵の組 (c, sk_{PO})，および，暗号文と時刻の組 (c, T) を復号クエリとして発行することができる．前者を時間前復号クエリ，後者を指定時間復号クエリと呼ぶこととする．$\mathcal{CH}$ はそれぞれのクエリに対し，前者は $k \leftarrow \mathsf{TRKEM.Decap_{PO}}(\mathsf{prm}, usk, sk_{PO}, c)$ を，後者はまず，$s_T \leftarrow \mathsf{TRKEM.Ext}(\mathsf{prm}, msk, T)$ を実行し，それを用いて $k \leftarrow \mathsf{TRKEM.Decap_{TR}}(\mathsf{prm}, usk, s_T, c)$ を返す

Challenge. $\mathcal{A}$ は任意の時刻 T^* を選び，$\mathcal{CH}$ に送る．$\mathcal{CH}$ はランダムにコイン $b_C \leftarrow \{0,1\}$ を振り，$(c^*, k_0^*, sk_{PO}^*) \leftarrow \mathsf{TRKEM.Enc}(\mathsf{prm}, upk, T^*)$, $k_1^* \leftarrow \mathcal{K}$ を計算し，$(c^*, k_{b_C}^*, sk_{PO}^*)$ を $\mathcal{A}$ に渡す．

Phase 2. $\mathcal{A}$ は Phase 1 と同様に，任意の回数の復号クエリを発行することができる．ただし，$(c, sk_{PO}) = (c^*, sk_{PO}^*)$，$(c, T) = (c^*, T^*)$ となるクエリを発行することはできない．

Guess. $\mathcal{A}$ は $\mathcal{CH}$ の選んだ b_C の予測として b_A を出力する．$b_A = b_C$ のとき $\mathcal{A}$ はゲームに勝利したという．

ここで，$\mathcal{A}$ が得るアドバンテージを以下のように定義する．

$$\mathsf{Adv}^{\text{IND-TRPC-CCA}^{\text{KEM}}_{\text{TS}}}_{\Pi,\mathcal{A}} = |\Pr[b_A = b_C] - 1/2|$$

定義 3.6. 全ての多項式時間アルゴリズム $\mathcal{A}$ に対し，$\mathsf{Adv}^{\text{IND-TRPC-CCA}^{\text{KEM}}_{\text{TS}}}_{\Pi,\mathcal{A}}$ が無視できるほど小さい値であるとき，Π は $\mathit{IND\text{-}TRPC\text{-}CCA}^{KEM}_{TS}$ 安全であるという．

受信者に対する安全性

ユーザーの秘密鍵を持った受信者が，時間前復号鍵，もしくは時刻鍵を持たなければ，時間前に暗号文から平文の情報を得ることができないという安全性．以下のゲームで定義される．

Setup. $\mathcal{CH}$ は $\mathsf{TRKEM.Setup}(1^\kappa)$，および $\mathsf{TRKEM.UKG}(1^\kappa)$ を実行する．出力された prm, usk, upk を $\mathcal{A}$ に渡し，msk を保持しておく．

Phase 1. $\mathcal{A}$ は $\mathcal{CH}$ に対し，任意の回数，時刻 T を時刻鍵導出クエリとして $\mathcal{CH}$ に発行することができ，$\mathcal{CH}$ はそのクエリに対し，正しい時刻鍵 $s_T \leftarrow \mathsf{TRKEM.Ext}(\mathsf{prm}, \mathsf{msk}, T)$ を返す．

Challenge. $\mathcal{A}$ は時刻 T^* を選び，$\mathcal{CH}$ に送る．$\mathcal{CH}$ はランダムにコイン $b_C \in \{0,1\}$ を振り，$(c^*, k_0^*, sk_{PO}^*) \leftarrow \mathsf{TRKEM.Enc}(\mathsf{prm}, upk, T^*, m_{b_C})$, $k_1^* \leftarrow \mathcal{K}$ を計算し，$(c^*, k_{b_C}^*)$ を $\mathcal{A}$ に渡す．ここで選ぶ T^* は，Phase 1 で聞いたどの T よりも大きくなければならない．

Phase 2. $\mathcal{A}$ は Phase 1 と同様に，任意の回数の時間鍵導出クエリを発行することができる．ただし，$T \geq T^*$ となる T をクエリとして発行することはできない．

Guess. $\mathcal{A}$ は $\mathcal{CH}$ の選んだ b_C の予測として b_A を出力する．$b_A = b_C$ のとき $\mathcal{A}$ はゲームに勝利したという．

ここで，$\mathcal{A}$ が得るアドバンテージを以下のように定義する．

$$\mathsf{Adv}^{\text{IND-TRPC-CPA}^{\text{KEM}}_{\text{IS}}}_{\Pi,\mathcal{A}} = |\Pr[b_A = b_C] - 1/2|$$

定義 3.7. 全ての多項式時間アルゴリズム $\mathcal{A}$ に対し，$\mathsf{Adv}^{\text{IND-TRPC-CPA}^{\text{KEM}}_{\text{IS}}}_{\Pi,\mathcal{A}}$ が無視できるほど小さい値であるとき，Π は $\mathit{IND\text{-}TRPC\text{-}CPA}^{KEM}_{IS}$ 安全であるという．

Binding

送信者は，時刻 T に関して暗号化した暗号文 c について，時刻鍵 s_T を用いて $\mathsf{TRKEM.Decap_{PO}}$ で復号した結果と，時間前開封鍵 sk_{PO} を用いて $\mathsf{TRKEM.Decap_{TR}}$ で復号した結果が異なる平文が出力されるような暗号文を作成することができないという安全性．以下のゲームで定義される．

Setup. $\mathcal{CH}$ は $\mathsf{TRE.Setup}(1^\kappa)$，および $\mathsf{TRE.UKG}(1^\kappa)$ を実行する．出力された prm, upk を $\mathcal{A}$ に渡し，msk, usk を保持しておく．

Query. $\mathcal{A}$ は $\mathcal{CH}$ に対し，任意の回数，時間前復号クエリ (c, sk_{PO})，および指定時間復号クエリ (c, T) を発行することができ，$\mathcal{CH}$ はそれぞれのクエリに対し，TS に対する安全性のゲームと同様に正しい復号結果 $m \leftarrow \mathsf{TRKEM.Decap_{PO}}(\mathsf{prm}, usk, sk_{PO}, c), m \leftarrow \mathsf{TRKEM.Decap_{TR}}(\mathsf{prm}, usk, s_T, c)$ を返す．同様に，$\mathcal{A}$ は $\mathcal{CH}$ に対し，任意の回数，時刻 T を時刻鍵導出クエリとして $\mathcal{CH}$ に発行することができ，$\mathcal{CH}$ はそのクエリに対し，正しい時刻鍵 $s_T \leftarrow \mathsf{TRKEM.Ext}(\mathsf{prm}, \mathsf{msk}, T)$ を返す．

Output. $\mathcal{A}$ は (c^*, T^*, sk_{PO}^*) の組を出力する．

ここで，$\mathcal{A}$が得るアドバンテージを以下のように定義する．

$$\begin{aligned}\mathsf{Adv}^{\mathrm{BINDING}^{\mathrm{KEM}}}_{\Pi,\mathcal{A}} = \Pr[\perp \neq \mathsf{TRKEM.Decap}_{\mathsf{PO}}(\mathsf{prm}, usk, sk^*_{PO}, c^*) \\ \neq \mathsf{TRKEM.Decap}_{\mathsf{TR}}(\mathsf{prm}, usk, s^*_T, c^*) \neq \perp]\end{aligned}$$

定義 3.8. 全ての多項式時間アルゴリズム $\mathcal{A}$ に対し，$\mathsf{Adv}^{\mathrm{BINDING}^{\mathrm{KEM}}}_{\Pi,\mathcal{A}}$ が無視できるほど小さい値であるとき，Π は *Binding* の安全性を満たすという．

3.4　アプリケーションへの応用

Rivest らは [45] にて、時間前開封機能のない TRE についてのアプリケーションへの応用として、いくつかのものが挙げられている。例としては、電子オークションや、暗号鍵供託システム (キーエスクロウシステム)、文書の指定時間以降の開示や、給与システム、プレスリリースなどがある。この中には、時間前開封機能のない TRE を用いるより、TRE-PC を用いたほうがより適している場合も多くある。

例えば、電子オークションでは、通常の場合入札者は、入札時間が終わるまで自身の入札価格は封をしておくものである。しかし、もし入札者が競売者に対して、自身の入札価格を、入札時間終了前に確認したくなった場合に、通常の TRE では不可能であるが、TRE-PC であれば、競売者が時間前開封鍵を入札者に送ることで容易にその機能を達成することができる。

別の例として、文書の条件付捺印証書をあげる。多くの法律制度では、機密扱いの政府の情報は特定の期間が過ぎた後に明らかにされることが義務付けられている。TRE-PC を用いることで、情報を開示する責任を持った機関の公開鍵を用いて機密情報を暗号化し、この機能を達成することができる。ここで、元の機密文書を保存する必要がないこと、また、情報を最初に設定された時期よりも早く公開する必要があった際に、開示する機関に時間前開封鍵を送ることで、容易に情報を開示することができるようになるという点がメリットとなると考えられる。

Chapter 4　関連研究

本章では本研究の関連研究を紹介する．また，章末では，それぞれの方式を 7 つの項目に分けて分類し，表として表している．

4.1　TRE, TRE-PC に関する既存研究

RSW96[45]

Rivest, Shamir, Wagner は 1996 年に，Time-lock puzzle による TRE の構成手法を提案した．また，同論文にて，Time Server を用いた TRE の効率的な利用法についての議論も行っている．

Mao01[38]

Mao は 2001 年に，RSW96 を改良した，Time-lock puzzle による TRE の構成手法を提案している．この方式ではより効率的に，また，指定時刻に対して正確に Time-lock puzzle による TRE を作成することを可能としている．

BC05[21]

Blake, Chan は 2005 年に，バイリニアペアリングを用いた，拡張可能性が高く，受動的なサーバーを用いた，ユーザーの匿名性の保たれている TRE 手法の構成法の提案を行っている．TS はユーザーとは一切相互通信を行わないため，TS の負荷が少なくてすむ方式となっている．また，期限付きの公開鍵のアップデートや鍵の断絶化（キーインシュレーション）も可能となっている．

CLQ05[18]

Cathalo, Libert, Quisquater は 2005 年に，バイリニアマップを用いた，TS との相互通信を行わない効率的な TRE の構成法の提案を行っている．また，新しい安全性の定義 (Release time confidentiality) も行っている．この定義は，送信者がどの時刻についての暗号文を作成したのかが TS には分からない，という安全性の定義である．

CHS07[19]

Chalkias, Hristu-Varsakelis, Stephanides は 2007 年に，匿名かつサーバーとの相互通信の不要な TRE を BC05，CLQ05，HYL05 の改良を行うことで構成した．また，BC05，CLQ05，HYL05 との計算コストの比較を行った．この方式では，計算コスト，および，鍵を保存するためのコストを削減することに成功している．同時に，この方

式の安全性の証明も行われている．

COR99[27]

Crescenz, Ostrovsky, Rajagopalan は 1999 年に，Conditional Oblivious Transfer という新しいプロトコルを作成した．また，作成した新しいプロトコルと，頑強性を持つ暗号化方式を用いることで，リクエスト一つ当たりの計算コストを，時間変数のログスケールに抑えられるような TRE を作成することに成功している．

CS06[20]

Chalkias, Stephanides は 2006 年に，ハッシュ連鎖および S/Key システムを使用した，バイリニアペアリングを用いた TRE を構成した．この方式では，最新の時刻鍵と，再帰的に検証を行うために必要となる最初の鍵のみを保持しておけばよく，以前の鍵は最新の鍵を用いて計算することができる．この方式を用いることで，時間に関する秘密鍵をサーバーが保持しておくコストを大幅に減らすことが可能になった．

CRR08[24]

Chow, Roth, Rieffel は 2008 年に，より強力な安全性を持った CLE(Certificateless Encryption) を構成した．また，これ以前の構成で作った CLE からでは，十分な安全性を持たない TRE しか作ることができなかったのだが，この研究によって構成された CLE を用いて TRE を構成することで，スタンダードモデルで必要な安全性を満たした TRE の構成を行うことが可能となっている．

CY08[25]

Chow, Yiu は 2008 年に，CHS07 の手法に対する新しい脅威を発見し，その安全性を破った．また，CRR08 による TRE のスタンダードモデルによる構成を拡張し，スタンダードモデルの TRE-PC の構成を行った．この方式は，受信者以外には暗号が復号できるようになる時間がわからない，という機能も同時に持つものである．この研究では初めてスタンダードモデルにおける TRE-PC の構成を行っているが，具体的な数論の困難性に基づいて構成を行っている上に，決して一般的とは言えないような数論的仮定に基づいて安全性の証明が行われており，極めて一般的に構成することのできる本稿の TRE-PC の構成法との大きな差異となっている．

DT07[28]

Dent, Tang は 2007 年に，HYL05 で提唱された TRE-PC の安全性モデルの分析を行った．その結果，HYL05 の安全性では TRE-PC が満たすべき安全性を網羅しておらず，足りていないと考えられる安全性の定義 (送信者に対する安全性：Binding) を行った．また，提案した安全性モデルを前提とした，TRE-PC KEM の構成も行っている．この構成法は具体的数論の困難性に基づいた方式となっており，また，ランダムオラクルモデルの下でのみ安全性が証明されている．本稿では，この研究によって

提案された TRE-PC の安全性の定義に基づいた構成を行っている．

HCS07[34]

Hristu-Varsakelis, Chalkias, Stephanides は 2007 年に，BC05 で提案された TRE を基に新しい匿名 TRE を構成した．この構成法はバイリニアペアリングを用いたものであり，非常に小さい計算コスト，および通信コストで TRE を構成することに成功している．また，この論文以前のいくつかの方式と計算コスト，通信コストの比較も行っている．さらに，この方式を，複数の TS を用いた仕様にする際のことについても簡単にではあるが触れられている．

HYL05[35]

Hwang, Yum, Lee は 2005 年に，3.2 節で述べた，時間前開封機能の付いた TRE(TRE-PC) の概念を提案し，また，安全性モデルも提案した．ただし，この安全性モデルについては DT07 で新たなモデルが考案されている．ランダムオラクルモデルのもとで BDH 仮定に基づいた TRE-PC の構成，および安全性の証明も行っている．また，認証付き電子メールシステムのアプリケーションを TRE-PC を用いて構築する方式を提案し，過去の同様のシステムとの間で通信コストの比較を行っている．

KM99[37]

Kudo, Mathuria は 1999 年に，Coffey, Saidha[26] によって提案された公開鍵暗号系のプロトコルをもとに，彼らのロジックを Timed-Release の公開鍵暗号系のプロトコルに拡張した．また，その拡張したロジックを適用し，いくつかの安全性に関する分析および証明を行っている．

NAM05[41]

Nali, Adams, Miri は 2005 年に，暗号文の長さが短い Foward Secure な HIBE(fsHIBE) を提案している．また，fsHIBE を用いることで，効率的な HTIR(hierarchical time-based information release) を構成することに成功した．また，この HTIR を，fsHIBE でない他の HIBE をもとに作成した HTIR と効率性の比較も行っている．

OS08[51]

岡本，斎藤は 2008 年に，一般的な TRE の機能に加えて，受信者が指定時刻の署名済み時報を入手できない場合に，送信者から復元信号をもらう事で，指定時刻以降の署名済み時報を使っても暗号文を復号できる機能を持った公開鍵型 TRE の作成を行った．また，復号信号がない場合には指定時刻以外では復号が出来ないようになっている．

SOOI02[54]

繁富, 大塚, 小川, 今井は 2002 年に，期限付きの匿名貸し出しプログラムの構成を行っており，そのアプリケーションを構成するための一部品として TRE を用いている．TRE の応用の一つの表現だと考えられる．

OKC04[44]
CHKO06[22]
CHKO08[23]

これら 3 つの文献は同一グループによる研究結果である．Cheon, Hopper, Kim, Osipkov は 2004 年から 2008 年にかけて TRE を SKIE-OTRU(Strongly key-insulated encryption with Optimal Threshold and Random Access Key Updates) から一般的に作成できるということをまず証明している．また，TR-PKAE(Authenticated Timed-Release Encryption) と呼ばれる，機能付き TRE の提案を行っている．TRE のモデルそのままでは，送られてきた暗号文が，本当に正しい送信者から送られてきたものかの確証を得ることができないため，暗号文に電子署名を加えなければならないという問題がある．この問題を解消するために，送信者自身が暗号文に認証をつけることで，暗号文と送信者の関係を偽れないようにすることが可能となる認証機能付き TRE が TR-PKAE である．この機能を達成するために，TR-PKAE では送信者も秘密鍵・公開鍵を持っており，送信者は自身の秘密鍵を用いて署名のような形で暗号文と関連付ける．受信者は送信者の公開鍵を用いて暗号文を検証することで，暗号文が正しく送信者から送られたものであるということを確認することができるようになっている．

また，彼らは TR-PKAE の安全性の定義を行い，まずバイリニアマップを用いた具体的数論の困難性に基づいた構成を，そして CHKO07 では TR-PKAE を公開鍵暗号，ID ベース暗号，および OneTime 署名から一般的に構成する手法を紹介している．

YMF04[52]
YMF065[50]

これらは同一グループによる研究結果である．Yoshida, Mitsunari, Fujiwara は，TRE において，受信者が TS からの署名付きの時報を手に入れなかった場合の対処方法を提案している．これらの研究では，解除信号と呼ばれる，その直前の期間を復号時刻とする全ての暗号を復号できるような信号を導入し，TS が一定期間ごとに解除信号を公開する．このようにすることで，TS は全ての署名付きの時報を保存しておく必要がなくなり，署名付きの時報を保管，公開するコストを下げることができると考えられる．

FCI09[53]

古川, 崔, 今井は 2009 年に，CY08 によるスタンダードモデルの TRE-PC を，より弱い仮定に基づき構成した．この論文では，Kurosawa-Desmedt 暗号，Waters IBE, One Time 署名から構成する方法を述べている．また，この構成法をより単純化し，さらに弱い仮定に基づいた TRE-PC の構成を行っている．

4.1.1　分類

4.1 節で調査した文献を，7 つの項目によって分類した．項目の種類，およびその意義についてを以下に挙げる．

新規モデル/機能追加 TRE の発展として、TRE-PC や TR-PKAE といった新規モデルの研究は、TRE を実用する面においてより効果的に用いるために必要である。ただし、モデルとは言っても、ある意味では TRE に新たな機能が追加されたものとも言えるので、新規モデルと TRE に機能を追加したものをまとめて一つの項目として分類した。

計算コスト減少 TRE を実用化する際、その計算量がどの程度になるのかという点は非常に重要となってくる。本項目では、計算コストを減少させることに成功した文献を分類している。

保存コスト減少 TRE に限らず、暗号を用いる際に鍵の管理の問題は必須となる。TRE の場合、時刻鍵を管理する必要がある。本項目では、時刻鍵をより効率的に管理する、もしくは時刻鍵を運用する際の工夫で鍵の管理コストを下げたものを分類している。

一般的構成法 本項目では、既存の暗号技術 (IBE や CLE) から TRE を構成する手法を扱っている文献を分類している。一般的構成法が存在することにより、より容易に TRE を構成することが可能であると考えられる。

匿名性 TRE において、TS の存在は必須ではあるが、送信者/受信者の側から考えると、誰と誰が通信しているのかを知られたくないという要求がある。本項目では、ユーザーと TS との間の匿名性を持った TRE についての文献を分類している。

安全性の改良 暗号技術において安全性は非常に重要である。本項目では、TRE の安全性をより完全なものにするために、新たな安全性の定義を行ったり、それに基づいた TRE を構成した文献を分類している。

応用 TRE をアプリケーションとして用いている文献を分類している。実用する際の参考になると考えられる。

以上の分類に従って、4.1 節の結果を視覚化したものを表 4.1 に示す。

表 4.1: 分類表

	新規モデル/機能追加	計算コスト減少	保存コスト減少	一般的構成法	匿名性	安全性の改良	応用
[RSW96]	Time-Lock Puzzles						
[Mao01]						○	
[BC05]	匿名 TRE				○		
[CLQ05]		○			○	○	
[CHS07]		○	○				
[CS06]			○				
[CRR08]			○				
[CY08]				○		○	
[DT07]						○	
[COR99]		○					
[HCS07]		○			○		
[HYL05]	TRE-PC						○
[KM99]						○	
[NAM05]		○					
[OS08]	指定時刻後の復号						
[SOOI02]							○
[OKC04]	TR-PKAE			○			
[CHKO06]							
[CHKO07]							
[YMF04]	解除信号導入		○				
[YMF06]							
[FCI09]				○			
[NMKM09]				○			
[NMKM10]			○	○			

Chapter 5　TRE-PCの一般的構成法

本章では，本研究のでの提案手法である，TRE-PC の一般的構成法を示す．提案する署名方式は 3 つあり，2 つはスタンダードモデルの下で，必要とされる安全性を全て満たす TRE-PC を，残りの 1 つはランダムオラクルモデルの下で，必要とされる安全性を全て満たす TRE-PC KEM を構成している．

まず，1.3 節で，既存の構成法，および，一般的構成法の意義について触れる．そして 5.2 節，5.3 節および 5.4 で，PKE, IBE, OneTime 署名を用いた TRE-PC 方式，TB-KEM, IB-KEM, Encapsulation を用いた TRE-PC 方式，およびに PK-KEM, IB-KEM を用いた TRE-PC KEM 方式ついてそれぞれ述べる．

5.1　既存の構成法/一般的構成法の意義

3.2 でも述べたように，TRE-PC は Hwang らによって初めて提案され，具体的な構成が示された [35]．彼らの構成法は，ランダムオラクルモデルの下で，bilinear Diffie-Hellman (BDH) 仮定に基づいて TRE-PC に必要とされる安全性が証明されている．

その後にもいくつかの TRE-PC の構成法が提案されている．

Dent ら [28] は [35] の安全性の定義を見直し，自分たちで提案したの安全性モデルを用いて，ランダムオラクルモデルの下で，BDH 仮定に基づいて安全な構成法を示している．この際，それぞれ安全な TRE-PC KEM と DEM を組み合わせることで，安全な TRE-PC となることも示している．

また，Chow ら [25] はスタンダードモデルの下で安全な構成法を初めて提案した．これは，彼らが提案した modified decisional 3-party Diffie-Hellman (3-MDDH) 仮定や，the decisional 3-party Diffie-Hellman (3-DDH) 仮定に基づいて安全な構成法となっている．

これらの構成法は，全て具体的な数論的問題の困難性に基づく構成法となっている．そのため，新しく実装をしようとすると，TRE-PC 用に仕様やプログラムを初めから作らなければならないため，実装コストが非常に高くなってしまう．

一方，一般的構成法は，既存の暗号技術をブラックボックスとして扱い，別の暗号を構成する手法である (1.3 参照)．ブラックボックスとして用いる暗号技術が必要とされる安全性を満たしていさえすれば，無条件に安全な別の暗号を構成することができる．そのため，条件を満たしている暗号を実装したものが既に存在している環境であれば，それらを一般的構成法の通りに組み合わせるだけで安全な暗号を作成できる．よって，一般的構成法が存在することにより，実装コストを大幅に下げることができると考えられる．

$\mathsf{TRE.Setup}(1^\kappa)$: Output $(\mathsf{msk}, \mathsf{prm}) \leftarrow \mathsf{IBE.Setup}(1^\kappa)$ $\mathsf{TRE.Ext}(\mathsf{prm}, \mathsf{msk}, T)$: $d_T \leftarrow \mathsf{IBE.Ext}(\mathsf{prm}, \mathsf{msk}, T)$ Output $s_T \leftarrow (d_T, T)$ $\mathsf{TRE.UKG}(1^\kappa)$: Output $(usk, upk) \leftarrow \mathsf{PKE.KG}(1^\kappa)$ $\mathsf{TRE.Enc}(\mathsf{prm}, T, upk, m)$: $r_1 \leftarrow \mathcal{M}, r_2 \leftarrow m \oplus r_1, R_{IBE} \leftarrow \mathcal{R}_{IBE}$ $(SK, VK) \leftarrow \mathsf{SigKG}(1^\kappa)$ $c_1 \leftarrow \mathsf{PKE.Enc}(upk, (VK\|\|r_1))$ $c_2 \leftarrow \mathsf{IBE.Enc}(\mathsf{prm}, T, (VK\|\|r_2); R_{IBE})$ $\sigma \leftarrow \mathsf{Sign}(SK, (T\|\|c_1\|\|c_2))$ $CT = (VK, T, c_1, c_2, \sigma)$ $sk_{PO} = (r_2, R_{IBE})$ Output (CT, sk_{PO})	$\mathsf{TRE.Dec_{TR}}(\mathsf{prm}, usk, s_T, CT)$: $(VK, T, c_1, c_2, \sigma) \leftarrow CT$ $(d_T, T') \leftarrow s_T$ If $\mathsf{Verify}(VK, (T\|\|c_1\|\|c_2), \sigma) =$ `reject` or $T \neq T'$ then output $\perp$ $(VK'\|\|r_1) \leftarrow \mathsf{PKE.Dec}(usk, c_1)$ $(VK"\|\|r_2) \leftarrow \mathsf{IBE.Dec}(d_T, c_2)$ if $VK = VK' = VK"$ then output $m = r_1 \oplus r_2$, else output $\perp$ $\mathsf{TRE.Dec_{PO}}(\mathsf{prm}, usk, sk_{PO}, CT)$: $(VK, T, c_1, c_2, \sigma) \leftarrow CT$ $(r_2, R_{IBE}) \leftarrow sk_{PO}$ If $\mathsf{Verify}(VK, (T\|\|c_1\|\|c_2), \sigma) =$ `reject` then output $\perp$ $(VK'\|\|r_1) \leftarrow \mathsf{PKE.Dec}(usk, c_1)$ $(c_2' = \mathsf{IBE.Enc}(\mathsf{prm}, T, (VK\|\|r_2); R_{IBE})$ if $c_2 = c_2'$ and $VK = VK'$ then output $m = r_1 \oplus r_2$, else output $\perp$

図 5.1: PKE,IBE,OneTime 署名から成る TRE-PC の一般的構成法

5.2　PKE,IBE,OneTime 署名を用いた TRE-PC の一般的構成法 (提案手法 1)

提案方式 Γ は IND-CCA 安全な公開鍵暗号 Π，IND-ID-CPA 安全かつ乱数に関するターゲット衝突困難性をもつな ID ベース暗号 Π'，および強偽造不可な OneTime 署名 Σ を用いて構成する．構成法を図 5.1 に示す．興味深いことに，この構成法は [23] などで言及されているような，時間前開封機能のない通常の TRE の PKE，IBE の多重暗号化法による構成と本質的に同じであり，ゼロ知識証明などのテクニックを必要としない．

5.2.1　提案手法 1 が安全性を満たす直観的な説明

詳細な証明を行う前に，何故本構成法で TRE-PC の安全性を達成できるのかということの，直観的な説明を行う．図 5.2 に提案手法 1 のイメージ図を示す．本構成の特徴は，時間前復号鍵として (r_2, R_{IBE}) を用いるという点である．ここで，r_2 は平文を秘密分散したうちの 1 つ，R_{IBE} は r_2 を暗号化する際に用いる乱数となっている．つまり，時間前復号鍵は暗号文 CT の IBE 部分である c_2 の全ての情報が含まれていることになる．何故このようなことが可能であるかというと，平文の秘密分散による多重暗

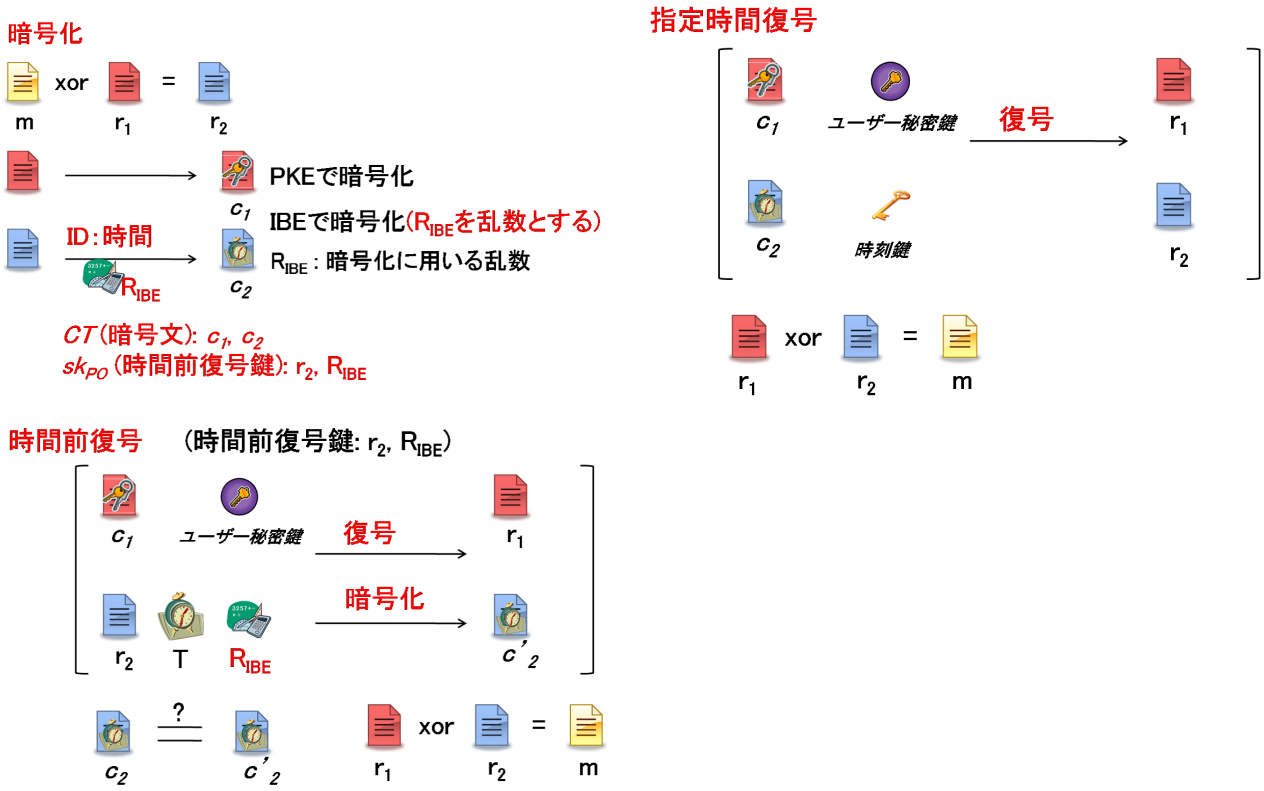

図 5.2: 提案手法 1 のイメージ図

号化を行っているため，r_2 のみからでは CT の平文の情報は一切得ることができないからである．また，R_{IBE} は，r_2 以上の平文の情報を漏らすこともありえない．そのため，攻撃者は，例え TRE-PC のマスター鍵を持っていたとしても，用いられている PKE の IND-CCA 安全性を破らなくては，受信者の秘密鍵なしに，(r_2, R_{IBE}) から平文の情報を得ることは不可能であり，このことが Time Server に対する安全性を満たす理由となっている．ただし，ある種の復号クエリに対し，用いる PKE の IND-CCA 安全性の力のみでは扱うことができないことが分かったため，その問題を解決するために，用いる IBE の乱数に関するもう 1 つの仮定を必要としている．この仮定は非常に妥当なものであると考えられる (2.2.2 参照)．

秘密鍵を持っており，なおかつ指定された時刻以前に平文の情報を得ようとする受信者に対する安全性は，用いる IBE の安全性によって達成される．これは，[23] にあるような，PKE および IBE からなる (時間前開封機能のない)TRE の一般的構成法と同様となっている．受信者は自分の秘密鍵 usk を持っているため，暗号文 CT の PKE 部分である c_1 に関する情報は全て得ることができる．しかし，上記と同様，平文の秘密分散による多重暗号化を行っているため，c_1 に含まれる平文の情報である r_1 のみからでは CT の平文の情報は一切得ることができない．そのため，指定時間以前に平文の情報を得ようとする受信者に対する安全性も達成することができる．

また，受信者が時間前復号鍵を用いて暗号文を復号する際には，受信者は時刻鍵 s_T を持っていないため，c_2 を実際に復号することはできない．しかし，時間前復号鍵

(r_2, R_{IBE}) を用いて c_2 を順方向に計算することで，暗号文の正当性のチェックを行い，もし受け取った暗号文と異なっていれば reject することができる．そのため，Binding の性質も満たすことができる．

また，c_1, c_2 の平文に含まれる署名鍵 VK は，暗号文同士を結び付けるセッション ID のように働いている．つまり，用いられている PKE, IBE で作られたそれぞれの暗号文を，TRE-PC の一つの暗号文として強く結び付け，もし間違った形の暗号文であれば reject することを可能としている．

興味深いことにこの構成法は，[23] で提案されている，(時間前開封機能のない)TRE の一般的構成法と，本質的には全く同じ構成をしており，特別なコストをかけることなく TRE-PC を構成することに成功している．そのため，もし [23] の一般的構成法で構成された TRE が存在すれば，その TRE から TRE-PC を構成することは容易である．

5.2.2　提案手法 1 の証明

本節では，提案手法 1 における，3.2.2 節で述べた 3 つの安全性についての詳細な証明を行う．

IND-TRPC-CCA$_{TS}$

定理 5.1. 構成要素である公開鍵暗号 Π が *IND-CCA* 安全性，*One-Time* 署名 Σ が *SUF-OTsig* 安全性，および，*ID* ベース暗号 Π' が乱数に関するターゲット衝突困難性を持つとき，提案方式 Γ は *IND-TRPC-CCA*$_{TS}$ の安全性を満たす．

証明　$\mathcal{A}$ を Γ の IND-TRPC-CCA$_{TS}$ ゲームに $\frac{1}{2} + \mathsf{Adv}^{\text{IND-TRPC-CCA}_{\text{TS}}}_{\Gamma,\mathcal{A}}$ の確率で勝利する攻撃者とし，$\mathcal{A}$ を用いて構成要素である Π の IND-CCA 安全性を破るシミュレータ $\mathcal{S}$ を構成する．矛盾を導くために，$\mathsf{Adv}^{\text{IND-TRPC-CCA}_{\text{TS}}}_{\Gamma,\mathcal{A}}$ を無視できないと仮定する．$\mathcal{S}$ は $\mathcal{A}$ に対して以下のように IND-TRPC-CCA$_{TS}$ ゲームのシミュレートを行いつつ利用して自身の IND-CCA チャレンジャー $\mathcal{CH}$ との間で以下のような IND-CCA ゲームを行う．

Setup. $\mathcal{S}$ は $\mathcal{CH}$ から upk を受け取る．また，$\mathsf{IBE.Setup}$ を実行し $(\mathsf{msk}, \mathsf{prm})$ を得る．$\mathcal{S}$ は $(\mathsf{msk}, \mathsf{prm}, upk)$ を $\mathcal{A}$ に渡す．

Phase 1. $\mathcal{S}$ は $\mathcal{A}$ の発行する時間前復号クエリ (CT, sk_{PO})，および指定時間復号クエリ (CT, T) に対して，以下のように応答する．前者については $\mathcal{S}$ は $\mathsf{TRE.Dec}_{\mathsf{PO}}$ に従い CT を復号し，m を返す．後者については，$\mathcal{S}$ はまず $\mathsf{IBE.Ext}(\mathsf{prm}, \mathsf{msk}, T)$ を計算し d_T を得る．そして，それを用いて $\mathsf{TRE.Dec}_{\mathsf{TR}}$ に従って CT を復号し，m を返す．どちらの復号の場合でも，$\mathcal{S}$ は $\mathsf{PKE.Dec}$ を自分自身で行う代わりに，$\mathcal{S}$ は自らの復号クエリとして c_1 を $\mathcal{CH}$ に発行し，$\mathcal{CH}$ から返ってきた値を用いる．

Challenge. $\mathcal{A}$ が (m_0, m_1, T^*) を出力したとき，$\mathcal{S}$ は以下のようにしてチャレンジ暗号文 CT^* を作成し，$\mathcal{A}$ に返す．まず，$\mathcal{S}$ は SigKG を実行し，(SK^*, VK^*) を作成

する．また，$r_2^* \leftarrow \mathcal{M}(m_0$ と長さの等しい乱数) および $R_{IBE}^* \leftarrow \mathcal{R}_{IBE}$ をそれぞれ一様ランダムに選び，$c_2^* \leftarrow \mathsf{IBE.Enc}(\mathsf{prm}, T^*, (VK^*||r_2^*); R_{IBE}^*)$ を計算する．ここで，$M_0 = (VK^*||m_0 \oplus r_2^*), M_1 = (VK^*||m_1 \oplus r_2^*)$ とする．(M_0, M_1)，および T^* を $\mathcal{S}$ 自身の Challenge として $\mathcal{CH}$ に対し出力し，c_1^* を $\mathcal{CH}$ から受け取る．その後，$\sigma^* \leftarrow \mathsf{Sign}(SK^*, (T^*||c_1^*||c_2^*))$ を計算する．最後に $CT^* = (VK^*, T^*, c_1^*, c_2^*, \sigma^*)$，および $sk_{PO}^* = (r_2^*, R_{IBE}^*)$ を $\mathcal{A}$ に渡す．

Phase 2. $\mathcal{S}$ は $\mathcal{A}$ の発行する時間前復号クエリ $(CT = (VK, T, c_1, c_2, \sigma), sk_{PO} = (r_2', R_{IBE}'))$ に対して以下のような応答を行う．ただし，ここで全て上から順に行うこととする．

(1) $\mathsf{Verify}(VK, (T||c_1||c_2), \sigma) = \mathtt{reject}$ **のとき:** $m \leftarrow \perp$ を返す．

(2) $c_1 \neq c_1^*$ **のとき:** Phase 1 と同様に $\mathsf{TRE.Dec_{PO}}$ を行い復号結果 m を返す．

(3) $c_1 = c_1^* \wedge VK = VK^* \wedge c_2 = \mathsf{IBE.Enc}(\mathsf{prm}, T^*, (VK||r_2'); R_{IBE}')$ **のとき:** シミュレーションを諦め停止する．

(4) それ以外のとき: 全て $m \leftarrow \perp$ を返す．

同様に，$\mathcal{S}$ は $\mathcal{A}$ の発行する指定時間復号クエリ $(CT = (VK, T, c_1, c_2, \sigma), T')$ に対して以下のような応答を行う．ここでも，上から順に行っていくこととする．

(1)' $\mathsf{Verify}(VK, (T||c_1||c_2), \sigma) = \mathtt{reject}$ **または** $T \neq T'$ **のとき:** $m \leftarrow \perp$ を返す．

(2)' $VK = VK^*$ **のとき:** シミュレーションを諦め停止する．

(3)' $c_1 = c_1^*$ **のとき:** $m \leftarrow \perp$ を返す．

(4)' それ以外のとき: Phase 1 と同様に $\mathsf{TRE.Dec_{TR}}$ を行い復号結果 m を返す．

Guess. $\mathcal{A}$ は b_A を出力する．$\mathcal{S}$ は b_A を自身の推測として出力する．

以上が $\mathcal{S}$ の構成である．ここで，以下のように 3 つのイベントを定義する．

Succ: 最終的に $\mathcal{S}$ が IND-CCA ゲームに勝利する．

$\mathsf{Abort_{TR}}$: Phase 2 において $\mathcal{A}$ が $\mathcal{S}$ を停止させるような指定時間復号クエリ，つまり，$\mathsf{Verify}(VK, (T||c_1||c_2), \sigma) = \mathtt{accept} \wedge VK = VK^* \wedge T' = T^*$ が成り立つようなクエリを少なくとも一つは発行する．

$\mathsf{Abort_{PO}}$: Phase 2 において $\mathcal{A}$ が $\mathcal{S}$ を停止させるような時間前復号クエリ，すなわち，$\mathsf{Verify}(VK, (T||c_1||c_2), \sigma) = \mathtt{accept} \wedge c_1 = c_1^* \wedge VK = VK^* \wedge c_2 = \mathsf{IBE.Enc}(\mathsf{prm}, T, (VK||r_2'); R_{IBE}')$ が成り立つようなクエリを少なくとも一つは発行する．

ここで，$\mathcal{S}$ がゲームに勝利する確率 $\Pr[\mathsf{Succ}]$ を計算すると以下のようになる．

$$\begin{aligned}
\Pr[\mathsf{Succ}] &\geq \Pr[\mathsf{Succ} \wedge \overline{\mathsf{Abort}_{\mathsf{PO}}} \wedge \overline{\mathsf{Abort}_{\mathsf{TR}}}] \\
&= \Pr[\mathsf{Succ}|\overline{\mathsf{Abort}_{\mathsf{PO}}} \wedge \overline{\mathsf{Abort}_{\mathsf{TR}}}] \cdot \Pr[\overline{\mathsf{Abort}_{\mathsf{PO}}} \wedge \overline{\mathsf{Abort}_{\mathsf{TR}}}] \\
&= \Pr[\mathsf{Succ}|\overline{\mathsf{Abort}_{\mathsf{PO}}} \wedge \overline{\mathsf{Abort}_{\mathsf{TR}}}] \cdot \Pr[\overline{\mathsf{Abort}_{\mathsf{PO}} \vee \mathsf{Abort}_{\mathsf{TR}}}] \\
&= \Pr[\mathsf{Succ}|\overline{\mathsf{Abort}_{\mathsf{PO}}} \wedge \overline{\mathsf{Abort}_{\mathsf{TR}}}] \cdot (1 - \Pr[\mathsf{Abort}_{\mathsf{PO}} \vee \mathsf{Abort}_{\mathsf{TR}}]) \\
&\geq \Pr[\mathsf{Succ}|\overline{\mathsf{Abort}_{\mathsf{PO}}} \wedge \overline{\mathsf{Abort}_{\mathsf{TR}}}] \cdot (1 - (\Pr[\mathsf{Abort}_{\mathsf{PO}}] + \Pr[\mathsf{Abort}_{\mathsf{TR}}])) \\
&\geq \Pr[\mathsf{Succ}|\overline{\mathsf{Abort}_{\mathsf{PO}}} \wedge \overline{\mathsf{Abort}_{\mathsf{TR}}}] - \Pr[\mathsf{Abort}_{\mathsf{PO}}] - \Pr[\mathsf{Abort}_{\mathsf{TR}}]
\end{aligned}$$

証明を完了するために以下の 3 つの補題を示す．

補題 1. $\Pr[\mathsf{Succ}|\overline{\mathsf{Abort}_{\mathsf{PO}}} \wedge \overline{\mathsf{Abort}_{\mathsf{TR}}}] = \frac{1}{2} + \mathsf{Adv}^{\text{IND-TRPC-CCA}_{\text{TS}}}_{\Gamma,\mathcal{A}}$.

補題 1 の証明　このゲームにおいて，$\mathcal{S}$ が $\mathcal{A}$ に与える鍵について，IND-TRPC-CCA$_{TS}$ ゲームの通りに正しくシミュレートが行われているのは明らかである．また，$\mathcal{S}$ は $\mathcal{A}$ にとってのチャレンジ暗号文を作成する際，構成法とは異なる作り方を行っている．我々の構成法では，r_2 は平文 m と一様ランダムに選んだ乱数 r_1 の排他的論理和をとって作成しているが，上記のゲームで $\mathcal{S}$ は，r_1 を平文 m と一様ランダムに選んだ乱数 r_2 の排他的論理和をとって作成している．しかし，乱数が正しく一様に分布しているならば，どちらの場合であっても全ての平文 m に対して，r_1 と r_2 の分布は完全に区別ができない．そのため，$\mathcal{A}$ に対するチャレンジは完璧にシミュレートできている．

ここで，$\mathsf{Abort}_{\mathsf{PO}}$，$\mathsf{Abort}_{\mathsf{TR}}$ が起こらない場合の復号クエリに対する $\mathcal{S}$ の応答のシミュレーションを考える．

(1) 時間前復号クエリに対する応答

$\mathsf{Verify}(VK, (T||c_1||c_2), \sigma) = \mathtt{reject}$ **のとき：**
$\mathcal{S}$ は $\perp$ を返す．これは明らかに正しくシミュレーションを行うことができている．

$c_1 \neq c_1^*$ **のとき：**
$\mathcal{S}$ は c_1 を復号クエリとして $\mathcal{CH}$ に問い合わせ，返された値を $\mathcal{A}$ に渡すことができる．このため，正しくシミュレーションを行うことが出来る．

$c_1 = c_1^* \wedge c_2 \neq \mathsf{IBE.Enc}(\mathsf{prm}, T, (VK||r_2'); R'_{IBE})$ **のとき：**
我々の構成法では，$c_2 \neq \mathsf{IBE.Enc}(\mathsf{prm}, T^*, (VK^*||r_2'); R'_{IBE})$ のとき $\perp$ を返す．つまり，$\perp$ を $\mathcal{A}$ に返すことで $\mathcal{S}$ は正しくシミュレーションを行うことができる．

$c_1 = c_1^* \wedge VK \neq VK^*$ **のとき：**
$c_1 = c_1^*$ であるので，c_1 の復号結果は $(VK^*||r_1^*)$ であり，$\mathcal{S}$ には r_1^* は未知な値である．しかし，仮定より $VK \neq VK^*$ であることから，この復号結果は矛盾するため，$\mathcal{S}$ は $\mathcal{A}$ に $\perp$ を返すことができる．このため，$\mathcal{S}$ は正しくシミュレーションを行うことができる．

(2) 指定時間復号クエリに対する応答

$\mathsf{Verify}(VK, (T||c_1||c_2), \sigma) = \mathtt{reject}$ **または** $T \neq T'$ **のとき:**

$\mathcal{S}$ は $\perp$ を返し，これは明らかに正しくシミュレートできている．$T \neq T'$ であるということは T' と関連する時刻鍵が T' そのものを含むということを意味しており，このとき，我々の指定時間復号アルゴリズム $\mathsf{TRE.Dec_{TR}}$ の最初のチェックを満たさない．(また，署名が reject されるということは当然 CT の復号結果が $\perp$ であることを意味する．)

$c_1 = c_1^*$ **のとき：**

$c_1 = c_1^*$ であるので，c_1 の復号結果は $(VK^*||r_1^*)$ であり，$\mathcal{S}$ には r_1^* は未知な値である．しかし，$VK \neq VK^*$ であることから ($VK = VK^*$ ならばシミュレーションを停止しているが，$\mathsf{Abort_{TR}}$ は起こっていないという仮定と矛盾する)，この復号結果は矛盾するため，$\mathcal{S}$ は $\mathcal{A}$ に $\perp$ を返すことができる．このため，$\mathcal{S}$ は正しくシミュレーションを行うことができる．

$c_1 \neq c_1^*$ **のとき：**

$\mathcal{S}$ は c_1 を復号クエリとして $\mathcal{CH}$ に問い合わせ，返された値を $\mathcal{A}$ に渡すことができる．このため，正しくシミュレーションを行うことが出来る．

以上により，$\mathsf{Abort_{PO}}$，$\mathsf{Abort_{TR}}$ が起こらないとき，$\mathcal{S}$ は $\mathcal{A}$ に対し，IND-TRPC-CCA_{TS} のゲームを完全にシミュレートしている．そのため，$\mathcal{S}$ の IND-CCA ゲームに勝利する確率は $\mathcal{A}$ が IND-TRPC-$\mathrm{CCA}_T S$ ゲームに勝利する確率と等しい．つまり，仮定より，

$$\Pr[\mathsf{Succ}|\overline{\mathsf{Abort_{PO}}} \wedge \overline{\mathsf{Abort_{TR}}}] = \frac{1}{2} + \mathsf{Adv}^{\text{IND-TRPC-CCA}_{\text{TS}}}_{\Gamma,\mathcal{A}}$$

が成り立つ．以上より，補題 1 は証明された．　□

補題 2. $\Pr[\mathsf{Abort_{TR}}]$ は無視できる値である．

補題 2 の証明　$\mathsf{Abort_{TR}}$ が起こるとき，条件により，$\mathcal{A}$ は $CT = (VK^*, T, c_1, c_2, \sigma) \wedge \mathsf{Verify}(VK^*, (T||c_1||c_2), \sigma) = \mathtt{accept}$ となる指定時間復号クエリ (CT, T) を最低一度は発行している (暗号文 CT に含まれる T とクエリに含まれる T は同一のものである)．また，$\mathcal{A}$ は IND-TRPC-CCA_{TS} 攻撃者なので，$(CT, T) \neq (CT^*, T^*)$ であり，この条件を満たす全てののクエリについて，$(T, c_1, c_2, \sigma) \neq (T^*, c_1^*, c_2^*, \sigma^*)$ が常に成り立つ．

ここで，$\Pr[\mathsf{Abort_{TR}}] = p_A$ で $\mathsf{Abort_{TR}}$ のイベントを起こす IND-TRPC-CCA_{TS} 攻撃者 $\mathcal{A}$ を用いて，構成要素である OneTime 署名 Σ の SUF-OTsig 安全性を破る攻撃者 $\mathcal{F}$ を以下のように構成する．ただし，p_A は無視できない値であるとする．

Setup. $\mathcal{F}$ は $\mathcal{CH}$ から VK^* を受け取る．また，$\mathsf{IBE.Setup}, \mathsf{PKE.KG}$ を実行し，$\mathcal{F}$ は $(\mathsf{msk}, \mathsf{prm})$，および，$(usk, upk)$ を得る．$\mathcal{F}$ は $(\mathsf{msk}, \mathsf{prm}, upk)$ を $\mathcal{A}$ に渡す．

Query. $\mathcal{F}$ は $\mathcal{A}$ の 2 種類の復号クエリに対して，msk と usk を持っているため，完全に応答することができる．$\mathcal{A}$ が (m_0, m_1, T^*) をチャレンジとして出力した場合，$R^*_{IBE} \leftarrow \mathcal{R}, r^*_2 \leftarrow \mathcal{M}, b_S \leftarrow \{0,1\}$ を一様ランダムに選び，$r^*_1 = m_{b_S} \oplus r_2$ を計算する．その後，$c^*_1 \leftarrow \mathsf{PKE.Enc}(upk, (VK^*||r^*_1))$，$c^*_2 \leftarrow \mathsf{IBE.Enc}(\mathsf{prm}, T^*, (VK^*||r^*_2); R^*_{IBE})$ を計算し，署名クエリとして $(T^*||c^*_1||c^*_2)$ を発行し，σ^* を得る．$\mathcal{F}$ はチャレンジ暗号文として $CT^* = (VK^*, T^*, c^*_1, c^*_2, \sigma^*)$，およびそれに対応する時間前復号鍵 $sk^*_{PO} = (r^*_2, R^*_{IBE})$ を $\mathcal{A}$ に渡す．

Forge. $\mathcal{A}$ が b_S の推測として b_A を出力した後，$\mathcal{F}$ は $\mathcal{A}$ の発行した指定時間復号クエリの中から，$VK = VK^* \wedge \mathsf{Verify}(VK^*, (T||c_1||c_2), \sigma) = \mathtt{accept} \wedge (T, c_1, c_2, \sigma) \neq (T^*, c^*_1, c^*_2, \sigma^*)$ となるものを探し，$((T||c_1||c_2), \sigma)$ を出力する．そのようなクエリがなければ停止する．

上記から明らかなように，$\mathcal{F}$ は，$\mathcal{A}$ が $\mathsf{Abort_{TR}}$ のイベントを起こす指定時間復号クエリを出すときには必ず偽造に成功する．つまり，

$$\mathsf{Adv}^{\text{SUF-OTsig}}_{\Sigma,\mathcal{F}} = \Pr[\mathsf{Abort_{TR}}] = p_A$$

となるが，これは p_A が無視できない値であるという仮定より，Σ が SUF-OTsig 安全であることと矛盾する．よって，$\Pr[\mathsf{Abort_{TR}}]$ は無視できる値である．以上より，補題 2 は証明された．　□

補題 3. $\Pr[\mathsf{Abort_{PO}}]$ は無視できる値である．

補題 3 の証明　$\mathsf{Abort_{PO}}$ が起こるとき，条件により，$\mathcal{A}$ は $CT = (VK^*, T, c^*_1, c_2, \sigma) \wedge c_2 = \mathsf{IBE.Enc}(\mathsf{prm}, T, (VK^*||r'_2); R'_{IBE}) \wedge \mathsf{Verify}(VK^*, (T^*||c^*_1||c_2), \sigma) = \mathtt{accept}$ となる時間前復号クエリ (CT, sk_{PO}) を最低一度は発行している．また，$\mathcal{A}$ が IND-TRPC-CCA$_{TS}$ 攻撃者なので，$(CT, sk_{PO}) \neq (CT^*, sk^*_{PO})$ であり，この条件を満たす全てのクエリについて，
(i)$(T, c_2, \sigma) \neq (T^*, c^*_2, \sigma^*)$
(ii)$(T, c_2, \sigma) = (T^*, c^*_2, \sigma^*) \wedge sk_{PO} = (r'_2, R'_{IBE}) \neq (r^*_2, R^*_{IBE})$
の 2 つの条件のうちどちらかが成り立ち，これ以外の可能性はない．ここで，$\mathsf{Abort_{PO}}$ のイベントを，Forge と Rand の 2 つのサブイベントに分ける．前者は $\mathcal{A}$ が (i) となるクエリを発行するというイベントであり，後者が (ii) となるクエリを発行するというイベントである．このとき，

$$\Pr[\mathsf{Abort_{PO}}] = \Pr[\mathsf{Forge}] + \Pr[\mathsf{Rand}]$$

が成り立つ．ここで，以下の 2 つについて考える．
(1)$\Pr[\mathsf{Forge}]$**：**
この値は用いる電子署名 Σ の SUF-OTsig 安全性により無視できる値である．証明は補題 2 とほぼ同じとなるため省略する．

(2)$\Pr[\mathsf{Rand}]$**：**

Rand が起こる時，$c_2^* = \mathsf{IBE.Enc}(\mathsf{prm}, T^*, (VK^*||r_2'); R_{IBE}')$ であるので，用いる IBE の正当性により $r_2' = r_2^*$ である．

ここで，確率 $\Pr[\mathsf{Rand}] = p_B$ で Rand のイベントを起こす IND-TRPC-CCA$_{TS}$ 攻撃者 $\mathcal{A}$ を想定し，構成要素である IBE 方式 Π' の乱数に関するターゲット衝突困難性を破る攻撃者 $\mathcal{H}$ を以下のように構成する．ただし，p_B は無視できない値であるとする．

Setup. $\mathcal{H}$ は $\mathcal{CH}$ から $\mathsf{msk}, \mathsf{prm}, R_{IBE}^*$ を受け取る．また，PKE.KG を実行し，$\mathcal{H}$ は (usk, upk) を得る．$\mathcal{H}$ は $(\mathsf{msk}, \mathsf{prm}, upk)$ を $\mathcal{A}$ に渡す．

Collision. $\mathcal{H}$ は $\mathcal{A}$ の 2 種類の復号クエリに対して，msk と usk を持っているため完全に応答できる．$\mathcal{A}$ が (m_0, m_1, T^*) をチャレンジとして出力した場合，$R_{IBE}^* \leftarrow \mathcal{R}, r_2^* \leftarrow \mathcal{M}, b_S \leftarrow \{0,1\}$ を一様ランダムに選び，$r_1^* = m_{b_S} \oplus r_2$ を計算する．また，$\mathcal{H}$ は SigKG を実行し，(SK^*, VK^*) を得る．次に $\mathcal{H}$ は，$c_1^* \leftarrow \mathsf{PKE.Enc}(pk_u, (VK^*||r_1^*))$，$c_2^* \leftarrow \mathsf{IBE.Enc}(\mathsf{prm}, T^*, (VK^*||r_2^*); R_{IBE}^*)$，および，$\sigma^* \leftarrow \mathsf{Sign}(SK^*, (T^*||c_1^*||c_2^*))$ を計算する．$\mathcal{H}$ はチャレンジ暗号文として $CT^* = (VK^*, T^*, c_1^*, c_2^*, \sigma^*)$，およびそれに対応する時間前復号鍵 $sk_{PO}^* = (r_2^*, R_{IBE}^*)$ を $\mathcal{A}$ に渡す．$\mathcal{A}$ が b_S の推測として b_A を出力した後，$\mathcal{H}$ は $\mathcal{A}$ の時間前復号クエリの中から Rand を起こすようなものを探し，$((VK^*||r_2^*), T^*, R_{IBE}')$ を出力する．そのようなクエリがなければ停止する．

上記から明らかなように，$\mathcal{H}$ は，$\mathcal{A}$ が Rand のイベントを起こす時間前復号クエリを発行するときには，必ず乱数に関するターゲット衝突困難性を破る．つまり，

$$\mathsf{Adv}_{\Pi',\mathcal{H}}^{\mathrm{Rand}} = \Pr[\mathsf{Rand}] = p_B$$

となるが，これは p_B が無視できない値であるという仮定より，Π' が乱数のターゲット衝突困難性を持つことと矛盾する．よって，$\Pr[\mathsf{Rand}]$ は無視できる値である．

(1)，(2) より，

$$\Pr[\mathsf{Abort}_{\mathsf{RK}}] = \Pr[\mathsf{Forge}] + \Pr[\mathsf{Rand}]$$

は無視できる値である．. 以上より，補題 3 は証明された．　□

補題 1，2，3 より，$\Pr[\mathsf{Succ}]$ を以下のように見積もることができる．

$$\Pr[\mathsf{Succ}] \geq \frac{1}{2} + \mathsf{Adv}_{\Gamma,\mathcal{A}}^{\text{IND-TRPC-CCA}_{\mathrm{TS}}} - \Pr[\mathsf{Abort}_{\mathsf{PO}}] - \Pr[\mathsf{Abort}_{\mathsf{TR}}].$$

ここで，$\mathsf{Adv}_{\Gamma,\mathcal{A}}^{\text{IND-TRPC-CCA}_{\mathrm{TS}}}$ は無視できない値であると仮定しているため，$\mathsf{Adv}_{\Pi,\mathcal{S}}^{\text{IND-CCA}} = |\Pr[\mathsf{Succ}] - \frac{1}{2}|$ も無視できない値となる．しかし，これは構成要素である PKE 方式 Π が IND-CCA 安全であるということと矛盾する．以上より，定理 5.1 は証明された．　□

IND-TRPC-CPA$_{IS}$

定理 5.2. 構成要素である Π' が *IND-ID-CPA* 安全な *ID* ベース暗号であるとき，提案方式 Γ は *IND-TRPC-CPA$_{IS}$* の安全性を満たす.

証明　$\mathcal{A}$ を IND-TRPC-CCA$_{IS}$ ゲームに $\frac{1}{2} + \mathsf{Adv}^{\text{IND-TRPC-CPA}_{\text{IS}}}_{\Gamma,\mathcal{A}}$ の確率で勝利する攻撃者とし，$\mathcal{A}$ を用いて構成要素である IBE 方式 Π' の IND-ID-CPA 安全性を破るシミュレータ $\mathcal{S}$ を構成する．矛盾を導くために，$\mathsf{Adv}^{\text{IND-TRPC-CPA}_{\text{IS}}}_{\Gamma,\mathcal{A}}$ を無視できない値であると仮定する．$\mathcal{S}$ は $\mathcal{A}$ に対して以下の様に IND-TRPC-CPA$_{IS}$ ゲームのシミュレートを行いつつ利用して自身の IND-ID-CPA チャレンジャー $\mathcal{CH}$ との間で以下のような IND-ID-CPA ゲームを行う.

Setup. $\mathcal{S}$ は $\mathcal{CH}$ から prm を受け取る．また，$\mathsf{PKE.KG}(1^\kappa)$ を実行し，(usk, upk) を得る．$\mathcal{S}$ は (prm, usk, upk) を $\mathcal{A}$ に渡す.

Phase 1. $\mathcal{S}$ は $\mathcal{A}$ の発行する時間鍵導出クエリ T に対して以下のように応答する．T を IBE における ID と考え，$\mathcal{CH}$ に鍵導出クエリとして問い合わせ，d_T を受け取る．$\mathcal{S}$ は $s_T \leftarrow (d_T, T)$ として，s_T を T に関する時刻鍵として $\mathcal{A}$ に渡す.

Challenge. $\mathcal{A}$ が (m_0, m_1, T^*) を出力したとき，$\mathcal{S}$ は以下の様にしてチャレンジ暗号文 CT^* を作成し，$\mathcal{A}$ に返す．まず，$\mathcal{S}$ は $(SK^*, VK^*) \leftarrow \mathsf{SigKG}$ を実行する．また，$r_1 \leftarrow \mathcal{M}$($m_0$ と長さの等しい乱数) を一様ランダムに選び，$c_1^* = \mathsf{PKE.Enc}(upk, (VK^*||r_1^*))$ を計算する．ここで，$M_0 = (VK^*||m_0 \oplus r_1^*)$，$M_1 = (VK^*||m_1 \oplus r_1^*)$ とする．(M_0, M_1)，および T^* を $\mathcal{S}$ 自身の Challenge として $\mathcal{CH}$ に対し出力し，c_2^* を $\mathcal{CH}$ から受け取る．その後，$\sigma^* \leftarrow \mathsf{Sign}(SK^*, (T^*||c_1^*||c_2^*))$ を計算する．最後に $CT^* = (VK^*, T^*, c_1^*, c_2^*, \sigma^*)$ を $\mathcal{A}$ に渡す.

Phase 2. Phase 1 と同様に時間鍵導出クエリに対して応答を行う.

Guess. $\mathcal{A}$ が b_A を出力する．$\mathcal{S}$ は b_A を自身の推測として出力する.

以上が $\mathcal{S}$ の構成である．ここで，$\mathcal{S}$ の $\mathcal{A}$ に対するシミュレーションは完全である．何故なら，$\mathcal{S}$ は正しいユーザー鍵の組 (usk, upk)，および，$\mathcal{CH}$ に対する鍵導出クエリを用いることで，Phase 1，Phase 2 における $\mathcal{A}$ が発行する時刻鍵導出クエリ T に対する正しい時刻鍵 s_T を $\mathcal{A}$ に渡しており，なおかつ $\mathcal{CH}$ から c_2^* を受け取り，それを用いてチャレンジ暗号文 CT^* を作成することで，$\mathcal{A}$ に完全に正しいチャレンジ暗号文を渡すことが出来ているためである．以上の結果より，$\mathcal{S}$ のアドバンテージは以下のように見積もることができる.

$$\mathsf{Adv}^{\text{IND-ID-CPA}}_{\Pi',\mathcal{S}} = \mathsf{Adv}^{\text{IND-TRPC-CPA}_{\text{IS}}}_{\Gamma,\mathcal{A}}$$

ここで仮定より，$\mathsf{Adv}^{\text{IND-TRPC-CPA}_{\text{IS}}}_{\Gamma,\mathcal{A}}$ は無視できない値である．しかし，これは構成要素である IBE 方式 Π' が IND-ID-CPA 安全であることと矛盾する．以上より，定理 5.2 は証明された.　□

Binding

定理 5.3. 提案方式である Γ は計算時間に縛られないあらゆる攻撃者に対して *Binding* の安全性を満たす.

証明　$CT = (VK, T, c_1, c_2, \sigma), sk_{PO} = (r_2, R_{IBE}), T'$ を Binding のゲームである攻撃者 $\mathcal{A}$ が出力したとする．このときの $\mathsf{Adv}^{\mathrm{BINDING}}_{\Gamma,\mathcal{A}}$ を計算すると以下のようになる．ただし，ここで $d_T \leftarrow \mathsf{IBE.Ext}(\mathsf{msk}, T), s_T \leftarrow (d_T, T)$ とする.

$$
\begin{aligned}
&\mathsf{Adv}^{\mathrm{BINDING}}_{\Gamma,\mathcal{A}} \\
&= \Pr[\mathsf{TRE.Dec_{PO}}(\mathsf{prm}, usk, sk_{PO}, CT) \neq \perp \wedge \mathsf{TRE.Dec_{TR}}(\mathsf{prm}, usk, s_T, CT) \neq \perp \\
&\qquad \wedge \mathsf{TRE.Dec_{PO}}(\mathsf{prm}, usk, sk_{PO}, CT) \neq \mathsf{TRE.Dec_{TR}}(\mathsf{prm}, usk, s_T, CT)] \\
&= \Pr[\mathsf{Verify}(VK, (T||c_1||c_2), \sigma) = \mathtt{accept} \wedge \mathsf{PKE.Dec}(usk, c_1) = (VK||r_1) \\
&\qquad \wedge c_2 = \mathsf{IBE.Enc}(\mathsf{prm}, T, (VK||r_2); R_{IBE}) \wedge \mathsf{IBE.Dec}(d_T, c_2) = (VK||r_2') \\
&\qquad \wedge r_1 \oplus r_2 \neq r_1 \oplus r_2'] \\
&\leq \Pr[c_2 = \mathsf{IBE.Enc}(\mathsf{prm}, T, (VK||r_2); R_{IBE}) \wedge \mathsf{IBE.Dec}(d_T, c_2) = (VK||r_2') \wedge r_2 \neq r_2'] \\
&= \Pr[r_2 = r_2' \wedge r_2 \neq r_2']
\end{aligned}
$$

ここで，最後の等式は構成要素である IBE 方式の正当性のためである．明らかに，$r_2 = r_2' \wedge r_2 \neq r_2'$ が成り立つことはないので，(計算時間に縛られない) あらゆる攻撃者に対して，$\mathsf{Adv}^{\mathrm{BINDING}}_{\Gamma,\mathcal{A}}$ は 0 となる．以上より，定理 5.3 は証明された．　□

5.3 TB-KEM, IB-KEM, Encapsulation を用いた TRE-PC の効率的な一般的構成法 (提案手法 2)

提案方式 Γ は IND-stag-wCCA 安全な TB-KEM 方式 Π，IND-ID-CPA 安全かつ乱数に関するターゲット衝突困難性をもつな IB-KEM 方式 Π'，hiding かつ binding な Encapsulation 方式 Υ, IND-COA 安全な共通鍵暗号方式 Λ, および SUF-OT$_{MAC}$ 安全な MAC 方式 Σ を用いて構成する．構成法を図 5.3 に示す.

5.3.1 提案手法 2 のアイデア

提案手法 2 のイメージ図を図 5.4 に示す．[16] において，Canetti，Halevi，及び Katz は強秘匿性を持つ IBE と One-time 署名を組み合わせて CCA 安全性を持つ PKE を構成する方法を提案した (CHK 変換と呼ぶ)．CHK 変換により得られる PKE の構成は非常に簡素で理解しやすいものであったが，One-time 署名の計算コスト及びパラメータのサイズによってもたらされる暗号文サイズの増加のために，実用的な PKE が得られるとは言い難かった.

CHK 変換は後に Boneh と Katz [13] によって大幅に改善された (BK 変換と呼ぶ)．BK 変換では，One-time 署名の代わりに，Encapsulation 方式および MAC 方式を用い

$\mathsf{TRE.Setup}(1^\kappa)$: $(\mathsf{msk}, \mathsf{prm_I}) \leftarrow \mathsf{IBKEM.Setup}(1^\kappa)$ $\mathsf{prm_E} \leftarrow \mathsf{ESetup}(1^\kappa)$ Output $(\mathsf{msk}, \mathsf{prm}) = (\mathsf{msk}, (\mathsf{prm_I}, \mathsf{prm_E}))$ $\mathsf{TRE.Ext}(\mathsf{prm}, \mathsf{msk}, T)$: $d_T \leftarrow \mathsf{IBKEM.Ext}(\mathsf{prm_I}, \mathsf{msk}, T)$ Output $s_T \leftarrow (d_T, T)$ $\mathsf{TRE.UKG}(1^\kappa)$: Output $(usk, upk) \leftarrow \mathsf{TBKEM.KG}(1^\kappa)$ $\mathsf{TRE.Enc}(\mathsf{prm}, T, upk, m)$: $(r, com, d) \leftarrow \mathsf{Com}(\mathsf{prm_E}), R_{IBE} \leftarrow \mathcal{R}_{IBE}$ $(c_1, k_1) \leftarrow \mathsf{TBKEM.Encap}(upk, com)$ $(c_2, k_2) \leftarrow \mathsf{IBKEM.Enc}(\mathsf{prm_I}, T; R_{IBE})$ $k \leftarrow k_1 \oplus k_2, c_3 \leftarrow \mathsf{SKE.Enc}(k, (m\|\|d))$ $\sigma \leftarrow \mathsf{MAC}(r, (c_1\|\|c_2\|\|c_3\|\|T))$ $CT = (com, c_1, c_2, c_3, T, \sigma)$ $sk_{PO} = R_{IBE}$ Output (CT, sk_{PO})	$\mathsf{TRE.Dec_{TR}}(\mathsf{prm}, usk, s_T, CT)$: $(com, c_1, c_2, c_3, T, \sigma) \leftarrow CT$ $(d_T, T') \leftarrow s_T$ $k_1 \leftarrow \mathsf{TBKEM.Decap}(usk, com, c_1)$ $k_2 \leftarrow \mathsf{IBKEM.Decap}(d_T, c_2)$ If $k_1 = \perp$ or $k_2 = \perp$ or $T \neq T'$ then output $\perp$ and stop. $k \leftarrow k_1 \oplus k_2, (m\|\|d) \leftarrow \mathsf{SKE.Dec}(k, c_3)$ $r \leftarrow \mathsf{Rec}(\mathsf{prm_E}, com, d)$ If $\mathsf{Verify_{MAC}}(r, (c_1\|\|c_2\|\|c_3\|\|T), \sigma) = \mathtt{accept}$ then output m, else output $\perp$ $\mathsf{TRE.Dec_{PO}}(\mathsf{prm}, usk, sk_{PO}, CT)$: $(com, c_1, c_2, c_3, \sigma) \leftarrow CT, R_{IBE} \leftarrow sk_{PO}$ $k_1 \leftarrow \mathsf{TBKEM.Decap}(usk, com, c_1)$ $(c_2', k_2) \leftarrow \mathsf{IBKEM.Encap}(\mathsf{prm}, T; R_{IBE})$ If $k_1 = \perp$ or $c_2 \neq c_2'$ then output $\perp$ and stop. $k \leftarrow k_1 \oplus k_2, (m\|\|d) \leftarrow \mathsf{SKE.Dec}(k, c_3)$ $r \leftarrow \mathsf{Rec}(\mathsf{prm_E}, com, d)$ If $\mathsf{Verify_{MAC}}(r, (c_1\|\|c_2\|\|c_3\|\|T), \sigma) = \mathtt{accept}$ then output m, else output $\perp$

図 5.3: TB-KEM,IB-KEM,Encapsulation から成る TRE-PC の一般的構成法

る．Encapsulation 方式は，ランダムな値をコミットするような特殊なコミットメント方式である (詳細は 2.8 節の定義を参照されたい)．この手法は，CHK 変換に比べ，変換後の PKE の暗号化，復号の際の計算コスト，及び暗号文サイズを大幅に改善する可能性を持つものであった．ただし，[13] において具体的に示された Encapsulation 方式は，やや大きなパラメータサイズを持っており，暗号文サイズの効率性の観点でいうとまだ改善の余地があった．そこで，Matsuda らは [39] において，特殊な仮定ではあるものの，現実的な仮定を用いることで Encapsulation 方式の効率を改善することに成功している．

本構成法では，ここで用いらている BK 変換のアイデアを用いて，OneTime 署名の代わりに Encapsulation 方式および MAC 方式を，また，KEM/DEM を用いることで，5.2 で提案した構成法の効率を，暗号文サイズの観点から良くすることができるのではないか，構成を行っている．

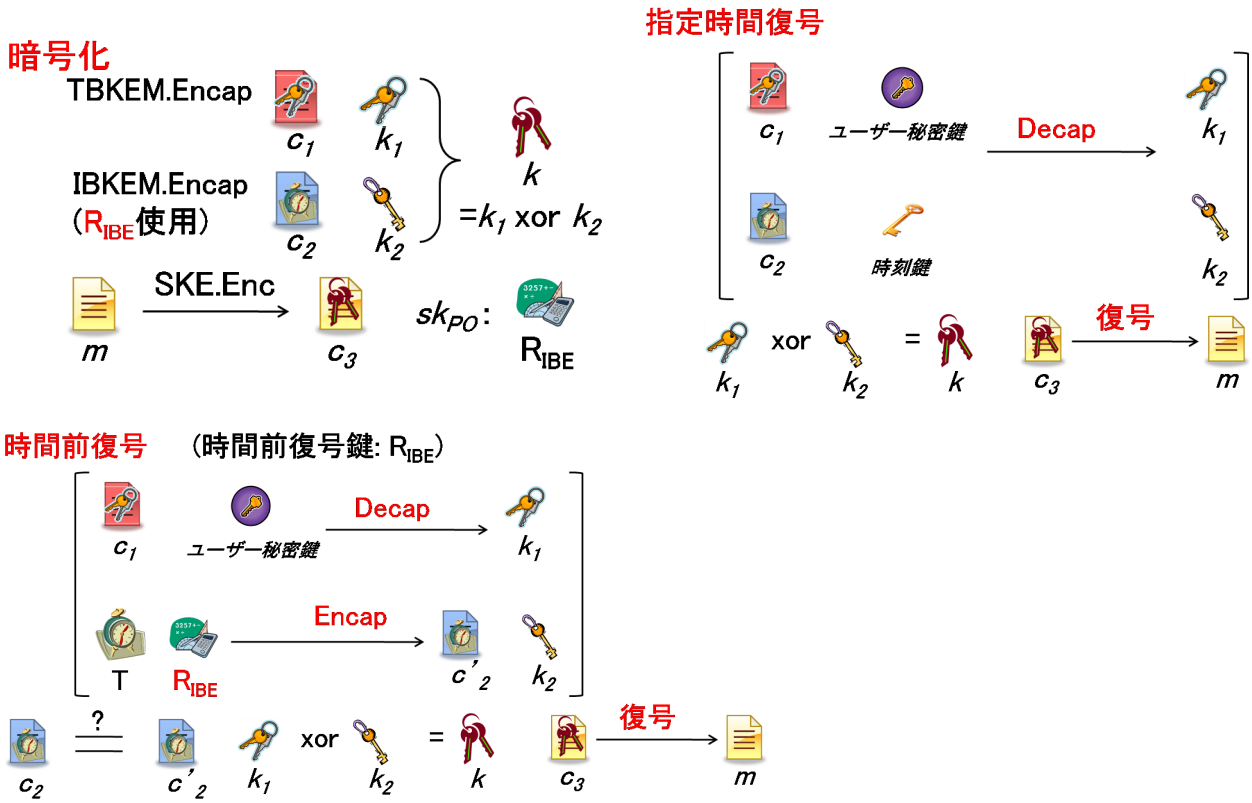

図 5.4: 提案手法 2 のイメージ図

5.3.2　提案手法 2 の証明

本節では，提案手法 2 における，3.2.2 節で述べた 3 つの安全性についての詳細な証明を行う．

IND-TRPC-CCA$_{TS}$

定理 5.4. 構成要素である *TB-KEM* 方式 Π が *IND-stag-wCCA* 安全性，共通鍵暗号 Λ が *IND-COA* 安全性，*Encapsulation* 方式 Υ が *hiding* かつ *binding* 安全性，*MAC* 方式 Σ が $SUF\text{-}OT_{MAC}$ 安全性，および，*IB-KEM* 方式 Π' が乱数に関するターゲット衝突困難性を持つとき，提案方式 Γ は $IND\text{-}TRPC\text{-}CCA_{TS}$ の安全性を満たす．

証明　$\mathcal{A}$ を，提案する TRE-PC 方式 Γ の IND-TRPC-CCA$_{TS}$ 安全性を破る攻撃者とすると，IND-TRPC-CCA$_{TS}$ ゲーム (3.2.2 参照) は以下のようになる．

Setup. $\mathcal{CH}$ は $\mathsf{TBKEM.KG}, \mathsf{IBKEM.Setup}, \mathsf{ESetup}, \mathsf{Com}(\mathsf{prm_E})$ を実行する．出力された $upk, \mathsf{msk}, \mathsf{prm_I}, \mathsf{prm_E}$ を $\mathcal{A}$ に渡し，$usk, (r^*, com^*, d^*)$ を保持しておく．

Phase 1. $\mathcal{A}$ は時間前復号クエリとして，$(CT = (com, c_1, c_2, c_3, T, \sigma), sk_{PO} = R_{IBE})$ を，指定時間復号クエリとして，$(CT = (com, c_1, c_2, c_3, T, \sigma), T)$ を，それぞれ任

意の回数発行することができる．$\mathcal{CH}$ は前者については，$\mathsf{TRE.Dec_{PO}}$ に従って復号した結果 $m/\perp$ を，後者については $d_T \leftarrow \mathsf{IBKEM.Ext}(\mathsf{prm}, \mathsf{msk}, T), s_T \leftarrow (d_T, T)$ を計算した後に，$\mathsf{TRE.Dec_{TR}}$ に従って復号した結果 $m/\perp$ を $\mathcal{A}$ に渡す．

Challenge. $\mathcal{A}$ は (m_0, m_1, T^*) をチャレンジとして出力する．$\mathcal{CH}$ はこれに対し，$R^*_{IBE} \leftarrow \mathcal{R}$, $(c^*_1, k^*_1) \leftarrow \mathsf{TBKEM.Encap}(upk, com^*)$, $(c^*_2, k^*_2) \leftarrow \mathsf{IBKEM.Encap}(\mathsf{prm}, T^*; R^*_{IBE})$, $k^* \leftarrow k^*_1 \oplus k^*_2$, $b_C \leftarrow \{0,1\}$, $c^*_3 = \mathsf{SKE.Enc}(k^*, (m_{b_C}||d^*), \sigma \leftarrow \mathsf{MAC}(r^*, (c^*_1||c^*_2||c^*_3||T^*))$ をそれぞれ順に実行し，チャレンジ暗号文 $CT^* = (com^*, c^*_1, c^*_2, c^*_{3b_C}, T^*, \sigma^*)$, およびそれに関連する時間前復号鍵 $sk_{PO} = R^*_{IBE}$ を $\mathcal{A}$ に渡す．

Phase 2. Phase 1 と同様に任意の回数の時間前復号クエリ，および指定時刻復号クエリを $\mathcal{A}$ は発行することができる．ただし，$(c, sk_{PO}) = (CT^* = (com^*, c^*_1, c^*_2, c^*_{3b_C}, T^*, \sigma^*), sk^*_{PO} = R^*_{IBE})$, $(CT, T) = (CT^* = (com^*, c^*_1, c^*_2, c^*_{3b_C}, T^*, \sigma^*), T^*)$ となるクエリを発行することはできない．

Guess. $\mathcal{A}$ は b_C の推測として b_A を出力する

さて，ここで以下のような 5 つのゲームを考える．

$\mathbf{Game}_0$: 上記の IND-TRPC-CCA$_{TS}$ ゲームを行う．

$\mathbf{Game}_1$: Game_0 において，Phase 2 の時間前復号クエリ，指定時間復号クエリに com^* を含む暗号文が投げられた場合は $\perp$ を返す．

$\mathbf{Game}_2$: Game_1 において，チャレンジの際に共通鍵 $k^* \leftarrow \mathcal{R}$(乱数) を用いる．

$\mathbf{Game}_3$: Game_2 において，チャレンジの際に $c_3 \leftarrow \mathsf{SKE.Enc}(k^*, 0^{|m_0|}||0^{|d^*|})$ を用いる．

$\mathbf{Game}_4$: Game_3 において，チャレンジの際にコミットされた値 $r^* \leftarrow \mathcal{R}$(乱数) を用いる．

また，以下の 4 つのイベントを定義する．

Succ_i**:** Game_i に $\mathcal{A}$ が勝つイベント

Valid_i**:** Game_i で $\mathcal{A}$ が，暗号文 CT の中に com^* を含み，かつ，復号結果が $\perp$ にならないような時間前復号クエリ，指定時間復号クエリを一度でも発行するイベント

Nobind_i**:** Game_i で $\mathcal{A}$ が，$com = com^*$ かつ $(m||d) \leftarrow \mathsf{SKE.Dec}(k, c_3), \mathsf{Rec}(\mathsf{prm_E}, com^*, d) \rightarrow r \notin \{r^*, \perp\}$ となる時間前復号クエリ，指定時間復号クエリを一度でも発行するイベント

Forge_i**:** Game_i で $\mathcal{A}$ が，$com = com^*$ かつ $\mathsf{Verify_{MAC}}(r^*, (c_1||c_2||c_3||T), \sigma) = \mathtt{accept}$ となる時間前復号クエリ，指定時間復号クエリを一度でも発行するイベント

このとき，$\mathcal{A}$ のアドバンテージは以下のように表すことができる．

$$\begin{aligned}\mathsf{Adv}^{\text{IND-TRPC-CCA}_{\text{TS}}}_{\Gamma,\mathcal{A}} &= |\Pr[\mathsf{Succ}_0] - \frac{1}{2}| \\ &\leq |\Pr[\mathsf{Succ}_0] - \Pr[\mathsf{Succ}_1]| + |\Pr[\mathsf{Succ}_1] - \frac{1}{2}| \\ &\leq \Pr[\mathsf{Valid}_1] + |\Pr[\mathsf{Succ}_1] - \frac{1}{2}|\end{aligned} \tag{5.1}$$

なぜなら，Game_1 において，Valid のイベントを起こすような復号クエリが一度も発行されなかった場合，Game_0 と Game_1 は全く同じゲームとなるため，$\Pr[\mathsf{Succ}_0]$ と $\Pr[\mathsf{Succ}_1]$ の差は，最大でも $\Pr[\mathsf{Valid}_1]$ となるからである．

ここで，さらに以下の数式が成り立つ．

$$\Pr[\mathsf{Valid}_1] \leq \Pr[\mathsf{Nobind}_1] + \Pr[\mathsf{Forge}_1] \tag{5.2}$$

これは，Valid が成り立つときは暗号文 CT の中に com^* を含んでおり，かつ復号結果が $\perp$ にならないのは上記の場合以外にありえないからである．

さて，ここで式 (5.1), (5.2) を元にさらに式を変形すると以下のように表すことができる．

$$\begin{aligned}\mathsf{Adv}^{\text{IND-TRPC-CCA}_{\text{TS}}}_{\Gamma,\mathcal{A}} &= |\Pr[\mathsf{Succ}_0] - \frac{1}{2}| \\ &\leq \Pr[\mathsf{Valid}_1] + |\Pr[\mathsf{Succ}_1] - \frac{1}{2}| \\ &\leq \Pr[\mathsf{Nobind}_1] + \Pr[\mathsf{Forge}_1] + |\Pr[\mathsf{Succ}_1] - \frac{1}{2}| \\ &\leq \Pr[\mathsf{Nobind}_1] + |\Pr[\mathsf{Forge}_1] - \Pr[\mathsf{Forge}_2]| + |\Pr[\mathsf{Forge}_2] - \Pr[\mathsf{Forge}_3]| \\ &\quad + |\Pr[\mathsf{Forge}_3] - \Pr[\mathsf{Forge}_4]| + \Pr[\mathsf{Forge}_4] + |\Pr[\mathsf{Succ}_1] - \Pr[\mathsf{Succ}_2]| \\ &\quad + |\Pr[\mathsf{Succ}_2] - \Pr[\mathsf{Succ}_3]| + |\Pr[\mathsf{Succ}_3] - \frac{1}{2}|\end{aligned} \tag{5.3}$$

この式について，以下の 6 つの補題を示す．

補題 4. $\Pr[\mathsf{Nobind}_1]$ は無視できる値である．

補題 4 の証明　$\Pr[\mathsf{Nobind}_1] = p_{bind}$ で Nobind_1 のイベントを起こす IND-TRPC-CCA$_{TS}$ 攻撃者 $\mathcal{A}$ を用いて，構成要素である Encapsulation 方式 Υ の binding 安全性を破る攻撃者 $\mathcal{F}$ を以下のように構成する．ただし，p_{bind} は無視できない値であるとする．

Setup. $\mathcal{B}$ は $\mathcal{CH}$ から $(\mathsf{prm}_\mathsf{E}, com^*, d^*)$ を受け取る．また，IBKEM.Setup, TBKEM.KG を実行し，$\mathcal{B}$ は $(\mathsf{msk}, \mathsf{prm}_\mathsf{I})$，および，$(usk, upk)$ を得る．$\mathcal{B}$ は $(\mathsf{msk}, \mathsf{prm}_\mathsf{I}, \mathsf{prm}_\mathsf{E}, upk)$ を $\mathcal{A}$ に渡す．

Query. $\mathcal{B}$ は $\mathcal{A}$ の 2 種類の復号クエリに対して，msk と usk を持っているため，完全に応答することができる．ただし，クエリ内の暗号文 CT に com^* が含まれている場合 $\perp$ を返す (Game_1 の定義と等しくするため)．$\mathcal{A}$ が (m_0, m_1, T^*) をチャレンジとして出力した場合，まず，$R^*_{IBE} \leftarrow \mathcal{R}$, $(c^*_1, k^*_1) \leftarrow \mathsf{TBKEM.Encap}(usk, com^*)$, $(c^*_2, k^*_2) \leftarrow \mathsf{IBKEM.Encap}(\mathsf{prm_I}, T^*; R^*_{IBE})$ を計算し，$k^* \leftarrow k^*_1 \oplus k^*_2$ を得る．次に，$b_B \leftarrow \{0,1\}$ を一様ランダムに選び，$c^*_3 = \mathsf{SKE.Enc}(k^*, m_{b_B}||d^*)$ を計算する．そして，$\mathsf{Rec}(\mathsf{prm_E}, com^*, d^*)$ を実行して r^* を得た後，$\mathsf{MAC}(r^*, (c^*_1||c^*_2||c^*_3||T^*))$ を実行し，σ^* を得る．$\mathcal{B}$ はチャレンジ暗号文として $CT^* = (com^*, c^*_1, c^*_2, c^*_3, T^*\sigma^*)$，およびそれに対応する時間前復号鍵 $sk^*_{PO} = (R^*_{IBE})$ を $\mathcal{A}$ に渡す．

Bind. $\mathcal{A}$ が b_B の推測として b_A を出力した後，$\mathcal{B}$ は $\mathcal{A}$ の発行した時間前復号クエリ，指定時間復号クエリの中から，$com = com^* \wedge (m||d) \leftarrow \mathsf{SKE.Dec}(k, c_3), \mathsf{Rec}(\mathsf{prm_E}, com^*, d) \rightarrow r \notin \{r^*, \perp\}$ となるものを探し，d を出力する．そのようなクエリがなければ停止する．

上記から明らかなように，$\mathcal{B}$ は，$\mathcal{A}$ が Nobind_1 のイベントを起こす復号クエリを出すときには必ず binding を破ることに成功する．つまり，

$$\mathsf{Adv}^{\mathrm{binding}}_{\Upsilon,\mathcal{B}} = \Pr[\mathsf{Nobind}_1] = p_{bind}$$

となるが，これは p_{bind} が無視できない値であるという仮定より，Υ が binding 安全であることと矛盾する．よって，$\Pr[\mathsf{Nobind}_1]$ は無視できる値である．以上より，補題 4 は証明された．　□

補題 5. $|\Pr[\mathsf{Succ}_1] - \Pr[\mathsf{Succ}_2]|$，および，$|\Pr[\mathsf{Forge}_1] - \Pr[\mathsf{Forge}_2]|$ は無視できる値である．

補題 5 の証明　IND-TRPC-CCA$_{TS}$ ゲームに対する攻撃者 $\mathcal{A}$ を用いて，構成要素である TB-KEM 方式 Π の IND-stag-wCCA 安全性を破る攻撃者 $\mathcal{TB}$ を以下のように構成する．また，$|\Pr[\mathsf{Succ}_1] - \Pr[\mathsf{Succ}_2] = p_{12}$ とし，p_{12} は無視できない値とする．

Setup. $\mathcal{TB}$ は $(r^*, com^*, d^*) \leftarrow \mathsf{Com}(\mathsf{prm_E})$ を実行し，攻撃対象となる com^* を選び，出力する．その後，$\mathcal{TB}$ は $\mathcal{CH}$ から upk を受け取る．また，$\mathsf{IBKEM.Setup}, \mathsf{ESetup}$ を実行し，$\mathsf{msk}, \mathsf{prm_I}, \mathsf{prm_E}$ を得る．$\mathcal{TB}$ は得られた $(\mathsf{msk}, \mathsf{prm_I}, \mathsf{prm_E}, upk)$ を $\mathcal{A}$ に渡す．

Phase 1. $\mathcal{TB}$ は $\mathcal{A}$ からの時間前復号クエリ，指定時間復号クエリに対して，クエリ内の暗号文 CT に com^* が含まれている場合 $\perp$ を返す (Game_1 の定義と等しくするため)．それ以外の場合，時間前復号クエリに関しては $\mathcal{TB}$ は $\mathsf{TRE.Dec_{PO}}$ に従い CT を復号し，m を返す．指定時間復号クエリについては，$\mathcal{TB}$ はまず $\mathsf{IBKEM.Ext}(\mathsf{prm_I}, \mathsf{msk}, T)$ を計算し d_T を得る．そして，それを用いて $\mathsf{TRE.Dec_{TR}}$ に従って CT を復号し，m を返す．どちらの復号の場合でも，$\mathcal{TB}$ は $\mathsf{TBKEM.Decap}$ を自分自身で行う代わりに，$\mathcal{TB}$ は自らの復号クエリとして c_1 を $\mathcal{CH}$ に発行し，$\mathcal{CH}$ から返ってきた値を用いる．

Challenge. $\mathcal{A}$ が (m_0, m_1, T^*) を出力したとき，$\mathcal{TB}$ は以下のようにしてチャレンジ暗号文 CT^* を作成し，$\mathcal{A}$ に返す．まず，$\mathcal{TB}$ はチャレンジ暗号文 $(c_1^*, k_{1b_C}^*)$ を受け取る．また，$R_{IBE}^* \leftarrow \mathcal{R}_{IBE}$ を一様ランダムに選んだ後，$(c_2^*, k_2^*) \leftarrow \mathsf{IBKEM.Encap}(\mathsf{prm}, T^*)$ を計算する．その後，$k^* \leftarrow k_{1b_C}^* \oplus k_2^*$ を得る．ここで，$\mathcal{TB}$ はランダムにコイン $\beta_{TB} \leftarrow \{0,1\}$ を振り，$c_{3\beta_{TB}}^* \leftarrow \mathsf{SKE.Enc}(k^*, (m_{\beta_{TB}}||d^*))$ を計算した後，$\sigma^* \leftarrow \mathsf{MAC}(r^*, (c_1^*||c_2^*||c_{3\beta_{TB}}^*||T^*))$ を計算する．最後に $\mathcal{TB}$ はチャレンジ暗号文として，$CT^* = (com^*, c_1^*, c_2^*, c_{3\beta_{TB}}^*, T^*, \sigma^*)$，および，$sk_{PO}^* = R_{IBE}^*$ を $\mathcal{A}$ に渡す．

Phase 2. Phase 1 と同様に任意の回数の時間前復号クエリ，指定時間復号クエリを $\mathcal{A}$ は発行することができる．

Guess. $\mathcal{A}$ は β_{TB} の推測として β_A を出力する．$\mathcal{TB}$ は $\beta_A = \beta_{TB}$ のとき b_C の推測として $b_{TB} = 0$ を，$\beta_A \neq \beta_{TB}$ のとき b_C の推測として $b_{TB} = 1$ を出力する．

このとき，$\mathcal{TB}$ のアドバンテージについて，以下のように計算できる．

$$\begin{aligned}
\mathsf{Adv}_{\Pi,\mathcal{TB}}^{\mathrm{IND-stag-wCCA}} &= |\Pr[b_{TB} = b_C] - \frac{1}{2}| \\
&= \frac{1}{2}|\Pr[b_{TB} = 0 | b_C = 0] - \Pr[b_{TB} = 0 | b_C = 1]| \\
&= \frac{1}{2}|\Pr[\beta_A = \beta_{TB} | b_C = 0] - \Pr[\beta_A = \beta_{TB} | b_C = 1]
\end{aligned} \tag{5.4}$$

ここで，最後の式について考えると，$b_C = 0$ のときのゲームは $\mathcal{A}$ にとっては Game_1 を正しくシミュレートしたものとなっている．つまり，

$$\Pr[\beta_A = \beta_{TB} | b_C = 0] = \Pr[\mathsf{Succ}_1] \tag{5.5}$$

であると言える．同様に，$b_C = 1$ のとき，$k^* \leftarrow k_{1b_C}^* \oplus k_2^*$ はランダムな値となっており，これは，$\mathcal{A}$ にとって，Game_2 が正しくシミュレートされたものとなっている．つまり，

$$\Pr[\beta_A = \beta_{TB} | b_C = 1] = \Pr[\mathsf{Succ}_2] \tag{5.6}$$

であると言える．

式 (5.4),(5.5),(5.6) より，

$$\begin{aligned}
\mathsf{Adv}_{\Pi,\mathcal{TB}}^{\mathrm{IND-stag-wCCA}} &= |\Pr[b_{TB} = b_C] - \frac{1}{2}| \\
&= \frac{1}{2}|\Pr[\mathsf{Succ}_1] - \Pr[\mathsf{Succ}_2]|
\end{aligned}$$

が成り立つ．つまり，$2 \times \mathsf{Adv}_{\Pi,\mathcal{TB}}^{\mathrm{IND-stag-wCCA}} = |\Pr[\mathsf{Succ}_1] - \Pr[\mathsf{Succ}_2]| = p_{12}$ となるが，仮定より p_{12} は無視できない値であり，これは Π が IND-stag-wCCA 安全であることと矛盾する．よって，$|\Pr[\mathsf{Succ}_1] - \Pr[\mathsf{Succ}_2]|$ は無視できる値である．

同様にして，$|\Pr[\mathsf{Forge}_1] - \Pr[\mathsf{Forge}_2]|$ も無視できる値となる．上記との違いは，Guess フェイズにおいて，$\mathcal{A}$ が β_{TB} の推測として β_A を出力した後，$\mathcal{A}$ の発行した復号クエリの中に $com = com^*$ かつ $\mathsf{Verify}_{\mathsf{MAC}}(r^*, (c_1||c_2||c_3||T), \sigma) = \mathtt{accept}$ となるクエリがあれば，$\mathcal{TB}$ は b_C の推測として $b_{TB} = 0$ を，それ以外の場合 b_C の推測として $b_{TB} = 1$ を出力する，という点である．

以上より，補題 5 は証明された．　□

補題 6. $|\Pr[\mathsf{Succ}_2] - \Pr[\mathsf{Succ}_3]|$，および，$|\Pr[\mathsf{Forge}_2] - \Pr[\mathsf{Forge}_3]|$ は無視できる値である．

補題 6 の証明　IND-TRPC-CCA$_{TS}$ ゲームに対する攻撃者 $\mathcal{A}$ を用いて，構成要素である SKE 方式 Λ の IND-COA 安全性を破る攻撃者 $\mathcal{SA}$ を以下のように構成する．また，$|\Pr[\mathsf{Succ}_2] - \Pr[\mathsf{Succ}_3] = p_{23}$ とし，p_{23} は無視できない値とする．

Setup. $\mathcal{SA}$ は $\mathsf{TBKEM.UKG}, \mathsf{IBKEM.Setup}, \mathsf{ESetup}$ を実行し，$(usk, upk), (\mathsf{msk}, \mathsf{prm_I}), \mathsf{prm_E}$ を得る．$\mathcal{SA}$ は $\mathcal{A}$ に $(\mathsf{msk}, \mathsf{prm_I}, \mathsf{prm_E}, upk)$ を渡す．

Phase 1. $\mathcal{SA}$ は $\mathcal{A}$ からの時間前復号クエリ，指定時間復号クエリに対して，クエリ内の暗号文 CT に com^* が含まれている場合 $\perp$ を返す (Game_1 の定義と等しくするため)．それ以外の場合，時間前復号クエリに関しては $\mathcal{SA}$ は $\mathsf{TRE.Dec_{PO}}$ に従い CT を復号し，m を返す．指定時間復号クエリについては，$\mathcal{SA}$ はまず $\mathsf{IBKEM.Ext}(\mathsf{prm_I}, \mathsf{msk}, T)$ を計算し d_T を得る．そして，それを用いて $\mathsf{TRE.Dec_{TR}}$ に従って CT を復号し，m を返す．これらは，$\mathcal{SA}$ が msk, usk を持っているため完全に応答することができる．

Challenge. $\mathcal{A}$ が (m_0, m_1, T^*) を出力したとき，$\mathcal{SA}$ は以下のようにしてチャレンジ暗号文 CT^* を作成し，$\mathcal{A}$ に返す．まず，$\mathcal{SA}$ は $(r^*, com^*, d^*) \leftarrow \mathsf{Com}(\mathsf{prm_E}), (c_1^*, k_1^*) \leftarrow \mathsf{TBKEM}.Encap(upk, com^*)$ を計算する．また，$R_{IBE}^* \leftarrow \mathcal{R}_{IBE}$ を一様ランダムに選んだ後，$(c_2^*, k_2^*) \leftarrow \mathsf{IBKEM.Encap}(\mathsf{prm}, T^*)$ を計算する．その後，$\mathcal{SA}$ はランダムにコイン $\beta_{SA} \leftarrow \{0,1\}$ を振り，$(m_{\beta_{SA}}||d^*), (0^{|m_0^*|}||0^n)$ を自身のチャレンジとして $\mathcal{CH}$ に送り，チャレンジ暗号文 $c_{3b_C}^*$ を受け取る．ただし，n は d^* の長さである．また，$\sigma^* \leftarrow \mathsf{MAC}(r^*, (c_1^*||c_2^*||c_{3b_C}^*||T^*))$ を計算する．最後に $\mathcal{SA}$ はチャレンジ暗号文として，$CT^* = (com^*, c_1^*, c_2^*, c_{3b_C}^*, T^*, \sigma^*)$，および，$sk_{PO}^* = R_{IBE}^*$ を $\mathcal{A}$ に渡す．

Phase 2. Phase 1 と同様に任意の回数の時間前復号クエリ，指定時間復号クエリを $\mathcal{A}$ は発行することができる．

Guess. $\mathcal{A}$ は β_{SA} の推測として β_A を出力する．$\mathcal{SA}$ は $\beta_A = \beta_{SA}$ のとき b_C の推測として $b_{SA} = 0$ を，$\beta_A \neq \beta_{SA}$ のとき b_C の推測として $b_{SA} = 1$ を出力する．

このとき，$\mathcal{SA}$ のアドバンテージについて，以下のように計算できる．

$$\begin{aligned}\mathsf{Adv}^{\mathrm{IND-COA}}_{\Lambda,\mathcal{SA}} &= |\Pr[b_{SA}=b_C]-\frac{1}{2}| \\ &= \frac{1}{2}|\Pr[b_{SA}=0|b_C=0]-\Pr[b_{SA}=0|b_C=1]| \\ &= \frac{1}{2}|\Pr[\beta_A=\beta_{SA}|b_C=0]-\Pr[\beta_A=\beta_{SA}|b_C=1]|\end{aligned} \tag{5.7}$$

補題 5 と同様に，$b_C=0$ のときは $\mathcal{CH}$ は正しいチャレンジ $m_{\beta_{SA}}$ を，$b_C=1$ のときは $0^{|m_0^*|}||0^n$ を暗号化して，$c^*_{3b_C}$ を計算し渡しているとしても一般性を失わない．すると，$b_C=0$ のときのゲームは $\mathcal{A}$ にとっては Game_2 を正しくシミュレートしたものに，$b_C=1$ のときのゲームは $\mathcal{A}$ にとっては Game_3 を正しくシミュレートしたものに他ならない．よって，

$$\Pr[\beta_A=\beta_{SA}|b_C=0]=\Pr[\mathsf{Succ}_2] \tag{5.8}$$

$$\Pr[\beta_A=\beta_{SA}|b_C=0]=\Pr[\mathsf{Succ}_3] \tag{5.9}$$

が成り立つ．

式 (5.7),(5.8),(5.9) より，

$$\begin{aligned}\mathsf{Adv}^{\mathrm{IND-COA}}_{\Lambda,\mathcal{SA}} &= |\Pr[b_{SA}=b_C]-\frac{1}{2}| \\ &= \frac{1}{2}|\Pr[\mathsf{Succ}_2]-\Pr[\mathsf{Succ}_3]|\end{aligned}$$

が成り立つ．つまり，$2\times\mathsf{Adv}^{\mathrm{IND-COA}}_{\Lambda,\mathcal{SA}}=|\Pr[\mathsf{Succ}_2]-\Pr[\mathsf{Succ}_3]|=p_{23}$ となるが，仮定より p_{23} は無視できない値であり，これは Λ が IND-COA 安全であることと矛盾する．よって，$|\Pr[\mathsf{Succ}_2]-\Pr[\mathsf{Succ}_3]|$ は無視できる値である．

同様にして，$|\Pr[\mathsf{Forge}_2]-\Pr[\mathsf{Forge}_3]|$ も無視できる値となる．上記との違いは，Guess フェイズにおいて，$\mathcal{A}$ が β_{SA} の推測として β_A を出力した後，$\mathcal{A}$ の発行した復号クエリの中に $com=com^*$ かつ $\mathsf{Verify}_{\mathsf{MAC}}(r^*,(c_1||c_2||c_3||T),\sigma)=\mathtt{accept}$ となるクエリがあれば，$\mathcal{SA}$ は b_C の推測として $b_{SA}=0$ を，それ以外の場合 b_C の推測として $b_{SA}=1$ を出力する，という点である．

以上より，補題 6 は証明された．　□

補題 7. $|\Pr[\mathsf{Forge}_3]-\Pr[\mathsf{Forge}_4]|$ は無視できる値である．

補題 7 の証明　IND-TRPC-CCA$_{TS}$ ゲームに対する攻撃者 $\mathcal{A}$ を用いて，構成要素である Encapsulation 方式 Υ の hiding 安全性を破る攻撃者 $\mathcal{HD}$ を以下のように構成する．また，$|\Pr[\mathsf{Forge}_3]-\Pr[\mathsf{Forge}_4]=p_{34}$ とし，p_{34} は無視できない値とする．

Setup. $\mathcal{HD}$ は $\mathcal{CH}$ から $(\mathsf{prm}_\mathsf{E},com^*,r^*_{b_C})$ を受け取る．また，IBKEM.Setup, TBKEM.KG を実行し，$(\mathsf{msk},\mathsf{prm}_\mathsf{I})$，および，$(usk,upk)$ を得る．$\mathcal{HD}$ は $(\mathsf{msk},\mathsf{prm}_\mathsf{I},\mathsf{prm}_\mathsf{E},upk)$ を $\mathcal{A}$ に渡す．

Hide. $\mathcal{HD}$ は $\mathcal{A}$ の 2 種類の復号クエリに対して，msk と usk を持っているため，完全に応答することができる．ただし，クエリ内の暗号文 CT に com^* が含まれている場合 $\perp$ を返す (Game_1 の定義と等しくするため)．$\mathcal{A}$ が (m_0, m_1, T^*) をチャレンジとして出力した場合，まず，$R^*_{IBE} \leftarrow \mathcal{R}$, $(c^*_1, k^*_1) \leftarrow \mathsf{TBKEM.Encap}(usk, com^*)$, $(c^*_2, k^*_2) \leftarrow \mathsf{IBKEM.Encap}(\mathsf{prm_I}, T^*; R^*_{IBE})$ を計算する．また，$k^* \leftarrow \mathcal{K}$ をランダムに選ぶ．次に，$\beta_{HD} \leftarrow \{0,1\}$ を一様ランダムに選び，$c^*_3 = \mathsf{SKE.Enc}(k^*, 0^{|m^*_0|}||0^n)$ を計算する．ただし，n は d の長さとする．そして，$\sigma^* \leftarrow \mathsf{MAC}(r^*_{b_C}, (c^*_1||c^*_2||c^*_3||T^*))$ を計算する．最後に $\mathcal{HD}$ はチャレンジ暗号文として，$CT^* = (com^*, c^*_1, c^*_2, c^*_3, T^*, \sigma^*)$，および，$sk^*_{PO} = R^*_{IBE}$ を $\mathcal{A}$ に渡す．$\mathcal{A}$ が自身のチャレンジの推測をした後，$\mathcal{HD}$ は $\mathcal{A}$ の発行した時間前復号クエリ指定時間復号クエリの中に，$com = com^*$ かつ $\mathsf{Verify}_{\mathsf{MAC}}(r^*_{b_C}, (c_1||c_2||c_3||T), \sigma) = \mathtt{accept}$ となるものがあれば b_C の推測として $b_{HD} = 0$ を，なければ b_C の推測として $b_{HD} = 1$ を出力する．

このとき，$\mathcal{HD}$ のアドバンテージについて，以下のように計算できる．

$$\begin{aligned}\mathsf{Adv}^{\mathrm{hiding}}_{\Upsilon,\mathcal{HD}} &= |\Pr[b_{HD} = b_C] - \frac{1}{2}| \\ &= \frac{1}{2}|\Pr[b_{HD} = 0 | b_C = 0] - \Pr[b_{HD} = 0 | b_C = 1]|\end{aligned}$$

ここで，補題 5 と同様に，$b_C = 0$ のときのゲームは $\mathcal{A}$ にとっては Game_3 を正しくシミュレートしたものに，$b_C = 1$ のときのゲームは $\mathcal{A}$ にとっては Game_4 を正しくシミュレートしたものに他ならない．

よって，

$$\begin{aligned}\mathsf{Adv}^{\mathrm{hiding}}_{\Upsilon,\mathcal{HD}} &= |\Pr[b_{HD} = b_C] - \frac{1}{2}| \\ &= \frac{1}{2}|\Pr[b_{HD} = 0 | b_C = 0] - \Pr[b_{HD} = 0 | b_C = 1]| \\ &= \frac{1}{2}|\Pr[\mathsf{Forge}_3] - \Pr[\mathsf{Forge}_4]|\end{aligned}$$

が成り立つ．つまり，$2 \times \mathsf{Adv}^{\mathrm{hiding}}_{\Upsilon,\mathcal{HD}} = |\Pr[\mathsf{Forge}_3] - \Pr[\mathsf{Forge}_4]| = p_{34}$ となるが，仮定より p_{34} は無視できない値であり，これは Υ が hiding 安全であることと矛盾する．よって，$|\Pr[\mathsf{Forge}_3] - \Pr[\mathsf{Forge}_4]|$ は無視できる値である．以上より，補題 7 は証明された．□

補題 8. $\Pr[\mathsf{Succ}_3] - \frac{1}{2} = 0$

補題 8 の証明　Game_3 において，チャレンジ暗号文は $0^{|m^*_0|}||0^{|d^*|}$ を暗号化したものとなっている．この暗号文にチャレンジとして送った m_0, m_1 の情報は一切入っていない．つまり，$\Pr[\mathsf{Succ}_3] = \frac{1}{2}$ となるのは明らかである．以上より，補題 8 は証明された．□

補題 9. $\Pr[\mathsf{Forge}_4]$ は無視できる値である．

補題 9 の証明　Forge_4 が起こる時，Game_4 において $\mathcal{A}$ は，$com = com^*$ かつ $\mathsf{Verify}_{\mathsf{MAC}}(r, (c_1||c_2||c_3||T), \sigma) = \mathtt{accept}$ となる指定時間復号クエリ，もしくは時間前復号クエリの少なくとも一方は確実に発行している．ここで，前者のクエリを $\mathsf{Abort}_{\mathsf{TR4}}$，後者のクエリを $\mathsf{Abort}_{\mathsf{PO4}}$ とすると，以下の式が成り立つ．

$$\Pr[\mathsf{Forge}_4] \leq \Pr[\mathsf{Abort}_{\mathsf{TR4}}] + \Pr[\mathsf{Abort}_{\mathsf{PO4}}] \tag{5.10}$$

(1)$\mathsf{Abort}_{\mathsf{TR4}}$ が起こる場合について：

このとき，条件により，$\mathcal{A}$ は $com = com^*$ かつ $\mathsf{Verify}_{\mathsf{MAC}}(r, (c_1||c_2||c_3||T), \sigma) = \mathtt{accept} \wedge T \neq T'$ となる指定時間復号クエリを最低一度は発行している．また，$\mathcal{A}$ が IND-TRPC-CCA$_{TS}$ 攻撃者なので，$(CT, T) \neq (CT^*, T^*)$ である．そのため，$CT = CT^* \wedge T \neq T^*$ とは成りえない (クエリの中の CT に含まれる T とクエリの中の T が等しくない場合には $\perp$ になるため)．つまり，常に $CT \neq CT^*$ が成り立つ．ここで，$\Pr[\mathsf{Abort}_{\mathsf{TR4}}] = p_{TR4}$ で $\mathsf{Abort}_{\mathsf{TR4}}$ のイベントを起こす IND-TRPC-CCA$_{TS}$ 攻撃者 $\mathcal{A}$ を用いて，構成要素である MAC 方式 Σ の SUF-OT$_{\mathsf{MAC}}$ 安全性を破る攻撃者 $\mathcal{F}$ を以下のように構成する．ただし，p_{TR4} は無視できない値であるとする．

Setup. $\mathcal{F}$ は $\mathsf{IBKEM.Setup}$, $\mathsf{TBKEM.KG}$, $\mathsf{ESetup}, \mathsf{Com}(\mathsf{prm_E})$ を実行し，$(\mathsf{msk}, \mathsf{prm_I})$, (usk, upk)，および $\mathsf{prm_E}, (r^*, com^*, d^*)$ を得る．$\mathcal{F}$ は $(\mathsf{msk}, \mathsf{prm_I}, \mathsf{prm_E}, upk)$ を $\mathcal{A}$ に渡す．

Query. $\mathcal{F}$ は $\mathcal{A}$ の 2 種類の復号クエリに対して，msk と usk を持っているため，完全に応答することができる．ただし，クエリ内の暗号文 CT に com^* が含まれている場合 $\perp$ を返す (Game_1 の定義と等しくするため)．$\mathcal{A}$ が (m_0, m_1, T^*) をチャレンジとして出力した場合，まず，$R^*_{IBE} \leftarrow \mathcal{R}$, $(r^*, com^*, d^*) \leftarrow \mathsf{Com}(\mathsf{prm_E})$, $(c^*_1, k^*_1) \leftarrow \mathsf{TBKEM.Encap}(usk, com^*)$, $(c^*_2, k^*_2) \leftarrow \mathsf{IBKEM.Encap}(\mathsf{prm_I}, T^*; R^*_{IBE})$ を計算する．$k^* \leftarrow \mathcal{K}$ をランダムに選ぶ．次に，$c^*_3 = \mathsf{SKE.Enc}(k^*, 0^{|m^*_0|}||0^n)$ を計算する．ただし，n は d の長さとする．そして，自身のクエリとして，$(c^*_1||c^*_2||c^*_3||T^*)$ を $\mathcal{CH}$ に送り，対応する MAC タグ σ^* を得る．最後に，$\mathcal{F}$ はチャレンジ暗号文として $CT^* = (com^*, c^*_1, c^*_2, c^*_3, T^*\sigma^*)$，およびそれに対応する時間前復号鍵 $sk^*_{PO} = R^*_{IBE}$ を $\mathcal{A}$ に渡す．

Forge. $\mathcal{A}$ が自身のチャレンジを推測した後，$\mathcal{F}$ は $\mathcal{A}$ の発行した指定時間復号クエリの中から，ランダムに一つを選び，$((c_1||c_2||c_3||T), \sigma)$ を出力する．そのようなクエリがなければ停止する．

$\mathcal{F}$ は，$\mathcal{A}$ が $\mathsf{Abort}_{\mathsf{TR4}}$ のイベントを起こす指定時間復号クエリを出すとき，ランダムに出力した $((c_1||c_2||c_3||T), \sigma)$ が偽造に成功している確率は，指定時間復号クエリの総数を q_{TR} とすると，$frac1q_{TR}$ となる．ここで，$\mathcal{A}$ が多項式時間アルゴリズムであるため，q_{TR} は高々多項式である．つまり，

$$\mathsf{Adv}^{\text{SUF-OT}_{\mathsf{MAC}}}_{\Sigma,\mathcal{F}} = \Pr[\mathsf{Abort}_{\mathsf{TR4}}] = \frac{p_{TR4}}{q_{TR}}$$

となるが，これは p_{TR4} が無視できない値であるという仮定より，$\frac{p_{TR4}}{q_{TR}}$ も無視できない値となり，これは Σ が $\text{SUF-OT}_{\mathsf{MAC}}$ 安全であることと矛盾する．よって，$\Pr[\mathsf{Abort}_{\mathsf{TR4}}]$ は無視できる値である．

(2)$\mathsf{Abort}_{\mathsf{PO4}}$ が起こる場合について：

このとき，条件により，$\mathcal{A}$ は $com = com^*$ かつ $(c_2, k_2) = \mathsf{IBKEM.Encap}(\mathsf{prm_I}, T; R'_{IBE}) \wedge \mathsf{Verify}(r_C, (c_1||c_2||c_3||T), \sigma) = \mathtt{accept}$ となる時間前復号クエリ (CT, sk_{PO}) を最低一度は発行している．また，$\mathcal{A}$ が IND-TRPC-CCA_{TS} 攻撃者なので，$(CT, sk_{PO}) \neq (CT^*, sk^*_{PO})$ である．この条件を満たす全てのクエリについて，

(i)$CT \neq CT^*$

(ii)$CT = CT^* \wedge sk_{PO} = R'_{IBE} \neq R^*_{IBE}$

の 2 つの条件のうちどちらかが成り立ち，これ以外の可能性はない．

ここで，$\mathsf{Abort}_{\mathsf{PO4}}$ のイベントを，MAC と Rand の 2 つのサブイベントに分ける．前者は $\mathcal{A}$ が (i) となるクエリを発行するというイベントであり，後者が (ii) となるクエリを発行するというイベントである．このとき，

$$\Pr[\mathsf{Abort}_{\mathsf{PO4}}] = \Pr[\mathsf{MAC}] + \Pr[\mathsf{Rand}]$$

が成り立つ．ここで，以下の 2 つについて考える．

(1)$\Pr[\mathsf{MAC}]$：

この値は用いる $\text{MAC}\Sigma$ の $\text{SUF-OT}_{\mathsf{MAC}}$ 安全性により無視できる値である．証明は上記とほぼ同じとなるため省略する．

(2)$\Pr[\mathsf{Rand}]$： Rand が起こる時，$(c^*_2, k^*_2) = \mathsf{IBKEM.Encap}(\mathsf{prm}, T^*; R'_{IBE})$ である．ここで，$\Pr[\mathsf{Abort}_{\mathsf{PO4}}] = p_{PO4}$ で $\mathsf{Abort}_{\mathsf{PO4}}$ のイベントを起こす IND-TRPC-CCA_{TS} 攻撃者 $\mathcal{A}$ を用いて，構成要素である IB-KEM 方式 Π' の乱数に関するターゲット衝突困難性を破る攻撃者 $\mathcal{H}$ を以下のように構成する．ただし，p_{rand} は無視できない値であるとする．

Setup. $\mathcal{H}$ は $\mathcal{CH}$ から $\mathsf{msk}, \mathsf{prm_I}, R^*_{IBE}$ を受け取る．また，$\mathsf{TBKEM.KG}, \mathsf{ESetup}$ を実行し，$\mathcal{H}$ は $(usk, upk), \mathsf{prm_E}$ を得る．$\mathcal{H}$ は $(\mathsf{msk}, \mathsf{prm}, upk)$ を $\mathcal{A}$ に渡す．

Collision. $\mathcal{H}$ は $\mathcal{A}$ の 2 種類の復号クエリに対して，msk と usk を持っているため完全に応答できる．$\mathcal{A}$ が (m_0, m_1, T^*) をチャレンジとして出力した場合，まず，$R^*_{IBE} \leftarrow \mathcal{R}$, $(r^*, com^*, d^*) \leftarrow \mathsf{Com}(\mathsf{prm_E})$, $(c^*_1, k^*_1) \leftarrow \mathsf{TBKEM.Encap}(usk, com^*)$, $(c^*_2, k^*_2) \leftarrow \mathsf{IBKEM.Encap}(\mathsf{prm_I}, T^*; R^*_{IBE})$ を計算する．また，$k^* \leftarrow \mathcal{K}$ をランダムに選び，$c^*_3 = \mathsf{SKE.Enc}(k^*, 0^{|m^*_0|}||0^n)$，および $\sigma^* \leftarrow \mathsf{MAC}(r^*_{b_C}, (c^*_1||c^*_2||c^*_3||T^*))$ を計算する．$\mathcal{H}$ はチャレンジ暗号文として $CT^* = (com^*, c^*_1, c^*_2, c^*_3, T^*\sigma^*)$，およびそれに対応する時間前復号鍵 $sk^*_{PO} = R^*_{IBE}$ を $\mathcal{A}$ に渡す．$\mathcal{A}$ が自身のチャレンジを推測した後，$\mathcal{H}$ は $\mathcal{A}$ の時間前復号クエリの中から Rand を起こすようなものを探し，(T^*, R'_{IBE}) を出力する．そのようなクエリがなければ停止する．

上記から明らかなように，$\mathcal{H}$ は，$\mathcal{A}$ が Rand のイベントを起こす時間前復号クエリを発行するときには，必ず乱数に関するターゲット衝突困難性を破る．つまり，

$$\mathsf{Adv}^{\mathrm{Rand}}_{\Pi',\mathcal{H}} = \Pr[\mathsf{Rand}] = p_{rand}$$

となるが，これは p_{rand} が無視できない値であるという仮定より，Π' が乱数のターゲット衝突困難性を持つことと矛盾する．よって，$\Pr[\mathsf{Rand}]$ は無視できる値である．

以上より

$$\Pr[\mathsf{Abort}_{\mathsf{PO}4}] = \Pr[\mathsf{MAC}] + \Pr[\mathsf{Rand}]$$

は無視できる値となる．

式 (5.10)，および (1)，(2) より，$\Pr[\mathsf{Forge}_4]$ は無視できる値となる．以上より，補題 9 は証明された． □

式 (5.3)，および補題 4，5，6，7，8，9 より，

$$\begin{aligned}\mathsf{Adv}^{\text{IND-TRPC-CCA}_{\mathrm{TS}}}_{\Gamma,\mathcal{A}} &= |\Pr[\mathsf{Succ}_0] - \frac{1}{2}| \\ &\leq \Pr[\mathsf{Nobind}_1] + |\Pr[\mathsf{Forge}_1] - \Pr[\mathsf{Forge}_2]| + |\Pr[\mathsf{Forge}_2] - \Pr[\mathsf{Forge}_3]| \\ &\quad + |\Pr[\mathsf{Forge}_3] - \Pr[\mathsf{Forge}_4]| + \Pr[\mathsf{Forge}_4] + |\Pr[\mathsf{Succ}_1] - \Pr[\mathsf{Succ}_2]| \\ &\quad + |\Pr[\mathsf{Succ}_2] - \Pr[\mathsf{Succ}_3]| + |\Pr[\mathsf{Succ}_3] - \frac{1}{2}|\end{aligned}$$

となり，$\mathsf{Adv}^{\text{IND-TRPC-CCA}_{\mathrm{TS}}}_{\Gamma,\mathcal{A}}$ は無視できる値となる．以上より，定理 5.4 は証明された． □

IND-TRPC-CPA$_{IS}$

定理 5.5. 構成要素である *IB-KEM* Π' が *IND-ID-CPA* 安全性，共通鍵暗号 Ω が *IND-COA* 安全性を持つとき，提案方式 Γ は *IND-TRPC-CPA*$_{IS}$ の安全性を満たす．

証明　$\mathcal{A}$ を，提案する TRE-PC 方式 Γ の IND-TRPC-CPA$_{IS}$ 安全性を破る攻撃者とすると，IND-TRPC-CPA$_{IS}$ ゲーム (3.2.2 参照) は以下のようになる．

Setup. $\mathcal{CH}$ は $\mathsf{TBKEM.KG}, \mathsf{IBKEM.Setup}, \mathsf{ESetup}$ を実行する．出力された $usk, upk, \mathsf{prm_I}, \mathsf{prm_E}$ を $\mathcal{A}$ に渡し，msk を保持しておく．

Phase 1. $\mathcal{A}$ は時刻鍵導出クエリとして T を任意の回数発行することができる．$\mathcal{CH}$ はこのクエリについて TRE.Ext に従って導出した時刻鍵 $s_T \leftarrow (d_T, T)$ を $\mathcal{A}$ に渡す．

Challenge. $\mathcal{A}$ は (m_0, m_1, T^*) をチャレンジとして出力する．$\mathcal{CH}$ はこれに対し，$(r^*, com^*, d^*) \leftarrow \mathsf{Com}(\mathsf{prm_E})$, $R^*_{IBE} \leftarrow \mathcal{R}$, $(c^*_1, k^*_1) \leftarrow \mathsf{TBKEM.Encap}(upk, com^*)$, $(c^*_2, k^*_2) \leftarrow \mathsf{IBKEM.Encap}(\mathsf{prm}, T^*; R^*_{IBE})$, $k^* \leftarrow k^*_1 \oplus k^*_2$, $b_C \leftarrow \{0,1\}$, $c^*_3 = \mathsf{SKE.Enc}(k^*, (m_{b_C}||d^*)$, $\sigma \leftarrow \mathsf{MAC}(r^*, (c^*_1||c^*_2||c^*_3||T^*))$ をそれぞれ順に実行し，チャレンジ暗号文 $CT^* = (com^*, c^*_1, c^*_2, c^*_{3b_C}, T^*, \sigma^*)$, およびそれに関連する時間前復号鍵 $sk_{PO} = R^*_{IBE}$ を $\mathcal{A}$ に渡す．

Phase 2. Phase 1 と同様に任意の回数の時刻鍵導出クエリを $\mathcal{A}$ は発行することができる．ただし，$T \geq T^*$ となるような T をクエリとして発行することはできない．

Guess. $\mathcal{A}$ は b_C の推測として b_A を出力する

さて，ここで以下のような 2 つのゲームを考える．

Game$_5$: 上記の IND-TRPC-CCA$_{IS}$ ゲームを行う．

Game$_6$: Game$_5$ において，チャレンジの際に共通鍵 $k^* \leftarrow \mathcal{R}$(乱数) を用いる．

それぞれのゲームで $\mathcal{A}$ が勝つイベントを Succ_i とすると，$\mathcal{A}$ のアドバンテージは以下のように表すことができる．

$$\begin{aligned} \mathsf{Adv}^{\text{IND-TRPC-CPA}_{\text{IS}}}_{\Gamma,\mathcal{A}} &= |\Pr[\mathsf{Succ}_5] - \frac{1}{2}| \\ &\leq |\Pr[\mathsf{Succ}_5] - \Pr[\mathsf{Succ}_6]| + |\Pr[\mathsf{Succ}_6] - \frac{1}{2}| \end{aligned} \tag{5.11}$$

この式について，以下の 2 つの補題を示す．

補題 10. $|\Pr[\mathsf{Succ}_5] - \Pr[\mathsf{Succ}_6]|$ は無視できる値である．

補題 10 の証明　IND-TRPC-CCA$_{IS}$ ゲームに対する攻撃者 $\mathcal{A}$ を用いて，構成要素である IB-KEM 方式 Π' の IND-ID-CPA 安全性を破る攻撃者 $\mathcal{IB}$ を以下のように構成する．また，$|\Pr[\mathsf{Succ}_5] - \Pr[\mathsf{Succ}_6]|$ とし，p_{56} は無視できない値とする．

Setup. $\mathcal{IB}$ は $\mathcal{CH}$ から $(\mathsf{msk}, \mathsf{prm}_\mathsf{I})$ を受け取る．また，$\mathsf{TBKEM.Setup}, \mathsf{ESetup}$ を実行し，$usk, \mathsf{prm}_\mathsf{E}$ を得る．$\mathcal{IB}$ は得られた $(usk, upk, \mathsf{prm}_\mathsf{I}, \mathsf{prm}_\mathsf{E})$ を $\mathcal{A}$ に渡す．

Phase 1. $\mathcal{IB}$ は $\mathcal{A}$ からの時刻鍵導出クエリについて，$\mathcal{IB}$ は $\mathsf{TRE.Ext}$ に従い T に対する時刻鍵を導出し，s_T を返す．この際，$\mathcal{IB}$ は $\mathsf{IBKEM.Ext}$ を自分自身で行う代わりに，$\mathcal{IB}$ は自らの鍵導出クエリとして T を $\mathcal{CH}$ に発行し，$\mathcal{CH}$ から返ってきた値を d_T とし，$s_T \leftarrow (d_T, T)$ を返す．

Challenge. $\mathcal{A}$ が (m_0, m_1, T^*) を出力したとき，$\mathcal{IB}$ は以下のようにしてチャレンジ暗号文 CT^* を作成し，$\mathcal{A}$ に返す．まず，$\mathcal{IB}$ は $(r^*, com^*, d^*) \leftarrow \mathsf{Com}$, $(c_1^*, k_1^*) \leftarrow \mathsf{TBKEM.Encap}(upk, com^*)$ を計算する．次に，$\mathcal{IB}$ は $\mathcal{CH}$ から自身のチャレンジ暗号文 $(c_2^*, k_{2b_C}^*)$ を受け取る．その後，$k^* \leftarrow k_1^* \oplus k_{2b_C}^*$ を得る．ここで，$\mathcal{IB}$ はランダムにコイン $\beta_{IB} \leftarrow \{0,1\}$ を振り，$c_{3\beta_{IB}}^* \leftarrow \mathsf{SKE.Enc}(k^*, (m_{\beta_{IB}}||d^*))$ を計算した後，$\sigma^* \leftarrow \mathsf{MAC}(r^*, (c_1^*||c_2^*||c_{3\beta_{IB}}^*||T^*))$ を計算する．最後に $\mathcal{IB}$ はチャレンジ暗号文として，$CT^* = (com^*, c_1^*, c_2^*, c_{3\beta_{IB}}^*, T^*, \sigma^*)$ を $\mathcal{A}$ に渡す．

Phase 2. Phase 1 と同様に任意の回数の時刻鍵復号クエリを $\mathcal{A}$ は発行することができる．

Guess. $\mathcal{A}$ は β_{IB} の推測として β_A を出力する．$\mathcal{IB}$ は $\beta_A = \beta_{IB}$ のとき b_C の推測として $b_{IB} = 0$ を，$\beta_A \neq \beta_{TB}$ のとき b_C の推測として $b_{IB} = 1$ を出力する．

このとき，$\mathcal{IB}$ のアドバンテージについて，以下のように計算できる．

$$\begin{aligned}\mathsf{Adv}^{\mathrm{IND-ID-CPA}}_{\Pi',\mathcal{IB}} &= |\Pr[b_{IB} = b_C] - \frac{1}{2}| \\ &= \frac{1}{2}|\Pr[b_{IB} = 0|b_C = 0] - \Pr[b_{IB} = 0|b_C = 1]| \\ &= \frac{1}{2}|\Pr[\beta_A = \beta_{IB}|b_C = 0] - \Pr[\beta_A = \beta_{IB}|b_C = 1]\end{aligned} \tag{5.12}$$

ここで，$b_C = 0$ のときのゲームは $\mathcal{A}$ にとっては Game_5 を正しくシミュレートしたものに，$b_C = 1$ のときのゲームは $\mathcal{A}$ にとっては Game_6 を正しくシミュレートしたものに他ならない．よって，

$$\Pr[\beta_A = \beta_{IB}|b_C = 0] = \Pr[\mathsf{Succ}_5] \tag{5.13}$$

$$\Pr[\beta_A = \beta_{IB}|b_C = 1] = \Pr[\mathsf{Succ}_6] \tag{5.14}$$

が成り立つ．

式 (5.12),(5.13),(5.14) より，

$$\begin{aligned}\mathsf{Adv}^{\mathrm{IND-ID-CPA}}_{\Pi',\mathcal{IB}} &= |\Pr[b_{IB} = b_C] - \frac{1}{2}| \\ &= \frac{1}{2}|\Pr[\mathsf{Succ5}] - \Pr[\mathsf{Succ}_6]|\end{aligned}$$

が成り立つ．つまり，$2 \times \mathsf{Adv}^{\mathrm{IND-ID-CPA}}_{\Pi',\mathcal{IB}} = |\Pr[\mathsf{Succ}_5] - \Pr[\mathsf{Succ}_6]| = p_{56}$ となるが，仮定より p_{56} は無視できない値であり，これは Π' が IND-ID-CPA 安全であることと矛盾する．よって，$|\Pr[\mathsf{Succ}_5] - \Pr[\mathsf{Succ}_6]|$ は無視できる値である．以上より，補題 10 は証明された．　□

補題 11. $|\Pr[\mathsf{Succ}_6] - \frac{1}{2}|$ は無視できる値である．

補題 11 の証明　$\mathcal{A}$ を IND-TRPC-CCA$_{TS}$ ゲームに $\frac{1}{2} + \mathsf{Adv}^{\text{IND-TRPC-CPA}_{\mathrm{IS}}}_{\Gamma,\mathcal{A}}$ の確率で勝利する攻撃者とし，$\mathcal{A}$ を用いて構成要素である SKE 方式 Λ の IND-COA 安全性を破る攻撃者 $\mathcal{SA}$ を以下のように構成する．矛盾を導くために，$|\Pr[\mathsf{Succ}_6] - \frac{1}{2}| = p_6$ を無視できない値であると仮定する．

Setup. $\mathcal{SA}$ は $\mathsf{TBKEM.UKG}, \mathsf{IBKEM.Setup}, \mathsf{ESetup}$ を実行し，$(usk, upk), (\mathsf{msk}, \mathsf{prm_I}), \mathsf{prm_E}$ を得る．$\mathcal{SA}$ は $\mathcal{A}$ に $(usk, upk, \mathsf{prm_I}, \mathsf{prm_E})$ を渡す．

Phase 1. $\mathcal{IB}$ は $\mathcal{A}$ からの時刻鍵導出クエリについて，$\mathcal{IB}$ は TRE.Ext に従い T に対する時刻鍵を導出し，s_T を返す．これは，$\mathcal{SA}$ が msk を持っているため完全に応答することができる．

Challenge. $\mathcal{A}$ が (m_0, m_1, T^*) を出力したとき，$\mathcal{SA}$ は以下のようにしてチャレンジ暗号文 CT^* を作成し，$\mathcal{A}$ に返す．まず，$\mathcal{SA}$ は $(r^*, com^*, d^*) \leftarrow \mathsf{Com}(\mathsf{prm}_{\mathsf{E}}), (c_1^*, k_1^*) \leftarrow \mathsf{TBKEM.Encap}(upk, com^*)$ を計算する．また，$R_{IBE}^* \leftarrow \mathcal{R}_{IBE}$ を一様ランダムに選んだ後，$(c_2^*, k_2^*) \leftarrow \mathsf{IBKEM.Encap}(\mathsf{prm}, T^*)$ を計算する．その後，$\mathcal{SA}$ は $(m_0||d^*), (m_1||d^*)$ を自身のチャレンジとして $\mathcal{CH}$ に送り，チャレンジ暗号文 $c_{3b_C}^*$ を受け取る．また，$\sigma^* \leftarrow \mathsf{MAC}(r^*, (c_1^*||c_2^*||c_{3b_C}^*||T^*))$ を計算する．最後に $\mathcal{SA}$ はチャレンジ暗号文として，$CT^* = (com^*, c_1^*, c_2^*, c_{3b_C}^*, T^*, \sigma^*)$ を $\mathcal{A}$ に渡す．

Phase 2. Phase 1 と同様に任意の回数の時間前復号クエリ，指定時間復号クエリを $\mathcal{A}$ は発行することができる．

Guess. $\mathcal{A}$ が b_A を出力する．$\mathcal{S}$ は b_A を自身の推測として出力する．

ここで，$\mathcal{SA}$ の $\mathcal{A}$ に対するシミュレーションは完全である．何故なら，$\mathcal{S}$ は正しいユーザー鍵の組 (usk, upk)，および，$\mathcal{CH}$ に対する鍵導出クエリを用いることで，Phase 1，Phase 2 における $\mathcal{A}$ が発行する時刻鍵導出クエリ T に対する正しい時刻鍵 s_T を $\mathcal{A}$ に渡しており，なおかつ $\mathcal{CH}$ から c_3^* を受け取り，それを用いてチャレンジ暗号文 CT^* を作成することで，$\mathcal{A}$ に完全に正しいチャレンジ暗号文を渡すことが出来ているためである．以上の結果より，$\mathcal{SA}$ のアドバンテージは以下のように見積もることができる．

$$\mathsf{Adv}_{\Lambda,\mathcal{SA}}^{\text{IND-COA}} = |\Pr[\mathsf{Succ}_6] - \frac{1}{2}| = p_6$$

ここで仮定より，p_6 は無視できない値である．しかし，これは構成要素である SKE 方式 Λ が IND-ID-CPA 安全であることと矛盾する．以上より，補題 11 は証明された．□

式 (5.11)，および補題 10，11 より，

$$\begin{aligned}\mathsf{Adv}_{\Gamma,\mathcal{A}}^{\text{IND-TRPC-CCA}_{\text{IS}}} &= |\Pr[\mathsf{Succ}_5] - \frac{1}{2}| \\ &\leq |\Pr[\mathsf{Succ}_5] - \Pr[\mathsf{Succ}_6]| + |\Pr[\mathsf{Succ}_6] - \frac{1}{2}|\end{aligned}$$

は無視できる値となる．以上より，定理 5.5 は証明された．□

Binding

定理 5.6. 提案方式である Γ は計算時間に縛られないあらゆる攻撃者に対して *Binding* の安全性を満たす．

証明　$CT = (com, c_1, c_2, c_3, T, \sigma), sk_{PO} = R_{IBE}, T'$ を Binding のゲームである攻撃者 $\mathcal{A}$ が出力したとする．このときの $\mathsf{Adv}^{\mathrm{BINDING}}_{\Gamma,\mathcal{A}}$ を計算すると以下のようになる．ただし，ここで $d_T \leftarrow \mathsf{IBKEM.Ext}(\mathsf{msk}, T), s_T \leftarrow (d_T, T)$ とする．

$$
\begin{aligned}
&\mathsf{Adv}^{\mathrm{BINDING}}_{\Gamma,\mathcal{A}} \\
&= \Pr[\mathsf{TRE.Dec_{PO}}(\mathsf{prm_I}, usk, sk_{PO}, CT) \neq \perp \wedge \mathsf{TRE.Dec_{TR}}(\mathsf{prm_I}, usk, s_T, CT) \neq \perp \\
&\qquad \wedge \mathsf{TRE.Dec_{PO}}(\mathsf{prm}, usk, sk_{PO}, CT) \neq \mathsf{TRE.Dec_{TR}}(\mathsf{prm}, usk, s_T, CT)] \\
&= \Pr[\mathsf{TBKEM.Decap}(usk, com, c_1) = k_1 \\
&\qquad \wedge (c_2, k_2) = \mathsf{IBKEM.Encap}(\mathsf{prm}, T; R_{IBE}) \wedge \mathsf{IBKEM.Decap}(d_T, c_2) = k_2' \qquad (5.15) \\
&\qquad \wedge \mathsf{SKE.Dec}(k_1 \oplus k_2, c_3) = (m||d) \wedge \mathsf{SKE.Dec}(k_1 \oplus k_2', c_3) = (m'||d') \\
&\qquad \wedge \mathsf{Rec}(\mathsf{prm_E}, com, d) = r \wedge \mathsf{Verify_{MAC}}(r, (c_1||c_2||c_3||T), \sigma) = \mathtt{accept} \\
&\qquad \wedge \mathsf{Rec}(\mathsf{prm_E}, com, d') = r' \wedge \mathsf{Verify_{MAC}}(r, (c_1||c_2||c_3||T), \sigma) = \mathtt{accept} \\
&\qquad \wedge m \neq m']
\end{aligned}
$$

ここで，IB-KEM の正当性より，

$$[(c_2, k_2) = \mathsf{IBKEM.Encap}(\mathsf{prm}, T; R_{IBE}) \wedge \mathsf{IBKEM.Decap}(d_T, c_2) = k_2'] = [k_2 = k_2']$$

が成り立つ．このとき，さらに SKE の正当性より，$k_2 = k_2'$ のとき，

$$[\mathsf{SKE.Dec}(k_1 \oplus k_2, c_3) = (m||d) \wedge \mathsf{SKE.Dec}(k_1 \oplus k_2', c_3) = (m'||d')] = [m = m']$$

つまり，式 5.16 は次と同値になる．

$$\mathsf{Adv}^{\mathrm{BINDING}}_{\Gamma,\mathcal{A}} = \Pr[m = m' \wedge m \neq m']$$

明らかに $m = m' \wedge m \neq m'$ が成り立つことはないので，(計算時間に縛られない) あらゆる攻撃者に対して $\mathsf{Adv}^{\mathrm{BINDING}}_{\Gamma,\mathcal{A}}$ は 0 となる．以上より，定理 5.6 は証明された．□

5.4　PK-KEM, IB-KEM を用いた TRE-PC KEM の効率的な一般的構成法 (提案手法 3)

提案方式 Γ は OW-CCA 安全な PK-KEM 方式 Π，OW-ID-CPA 安全かつ乱数に関するターゲット衝突困難性をもつな IB-KEM 方式 Π'，およびランダムオラクル H を用いて構成する．構成法を図 5.5 に示す．

5.4.1　提案手法 3 のアイデア

提案手法 3 のイメージ図を図 5.6 に示す．提案手法 3 では，PK-KEM，IB-KEM の 2 つの方式を用いて暗号文，および共通鍵を作成する方式となっている．提案手法 1，2 と同じように，PK-KEM 部 (c_1) に関してはユーザーの秘密鍵によって，IB-KEM 部 (c_2)

$\mathsf{TRE.Setup}(1^\kappa)$: Output $(\mathsf{msk}, \mathsf{prm}) \leftarrow \mathsf{IBKEM.Setup}(1^\kappa)$ $\mathsf{TRE.Ext}(\mathsf{prm}, \mathsf{msk}, T)$: $d_T \leftarrow \mathsf{IBKEM.Ext}(\mathsf{prm}, \mathsf{msk}, T)$ Output $s_T \leftarrow (d_T, T)$ $\mathsf{TRE.UKG}(1^\kappa)$: Output $(usk, upk) \leftarrow \mathsf{PKKEM.KG}(1^\kappa)$ $\mathsf{TRE.Encap}(\mathsf{prm}, T, upk, m)$: $R_{IBE} \leftarrow \mathcal{R}_{IBE}$ $(c_1, k_1) \leftarrow \mathsf{PKKEM.Encap}(upk)$ $(c_2, k_2) \leftarrow \mathsf{IBKEM.Enc}(\mathsf{prm}, T; R_{IBE})$ $CT = (c_1, c_2, T)$, $k \leftarrow H(c_1, c_2, k_1, k_2, T)$ $sk_{PO} = R_{IBE}$ Output (CT, k, sk_{PO})	$\mathsf{TRE.Decap_{TR}}(\mathsf{prm}, usk, s_T, CT)$: $(c_1, c_2, T) \leftarrow CT$ $k_1 \leftarrow \mathsf{PKKEM.Decap}(usk, c_1)$ $k_2 \leftarrow \mathsf{IBKEM.Decap}(d_T, c_2)$ If $k_1 = \perp$ or $k_2 = \perp$ then output $\perp$, else output $k \leftarrow H(c_1, c_2, k_1, k_2, T)$ $\mathsf{TRE.Decap_{PO}}(\mathsf{prm}, usk, sk_{PO}, CT)$: $(c_1, c_2, T) \leftarrow CT$, $R_{IBE} \leftarrow sk_{PO}$ $k_1 \leftarrow \mathsf{PKKEM.Decap}(usk, c_1)$ $(c_2', k_2) \leftarrow \mathsf{IBKEM.Encap}(\mathsf{prm}, T; R_{IBE})$ If $k_1 = \perp$ or $c_2 \neq c_2'$ then output $\perp$, else output $k \leftarrow H(c_1, c_2, k_1, k_2, T)$

図 5.5: PK-KEM, IB-KEM から成る TRE-PC KEM の一般的構成法

については TS から送られてくる時刻鍵によって暗号文を復号できるようになっている．また，それぞれの方式から作成される暗号文，およびその復号結果である鍵をすべてハッシュ$(H(c_1, c_2, k_1, k_2))$することによって，それぞれの暗号文およびその復号結果である共通鍵を強く結び付けている．ランダムオラクルモデルであるため，ハッシュする要素が 1bit でも異なれば，H の結果として全く異なる値が出力されることが保証されているため，それぞれの暗号文を正しく復号できなければ，正しい共通鍵を復号できないようになっている．

また，時間前復号についても提案手法 1，2 と同様に，IB-KEM の Encap アルゴリズムに用いる乱数 R_{IBE} を時間前復号鍵として用いる．これを用いることで，IB-KEM 部に関する情報 (c_2, k_2) は全て公開されてしまうことになるが，上で述べたように，実際に共通鍵を復号するためには (c_1, k_1) も必要となるため，(c_2, k_2) からのみでは正しい共通鍵を得ることができないようになっている．

5.4.2　提案手法 3 の証明

本節では，提案手法 3 における，3.3.2 節で述べた 3 つの安全性についての詳細な証明を行う．

IND-TRPC-CCA$^{KEM}_{TS}$ 安全性

定理 5.7. 構成要素である *PK-KEM* 方式 Π が *OW-CCA* 安全性，*IB-KEM* 方式 Π' が乱数に関するターゲット衝突困難性，および H がランダムオラクルとみなせるとき，

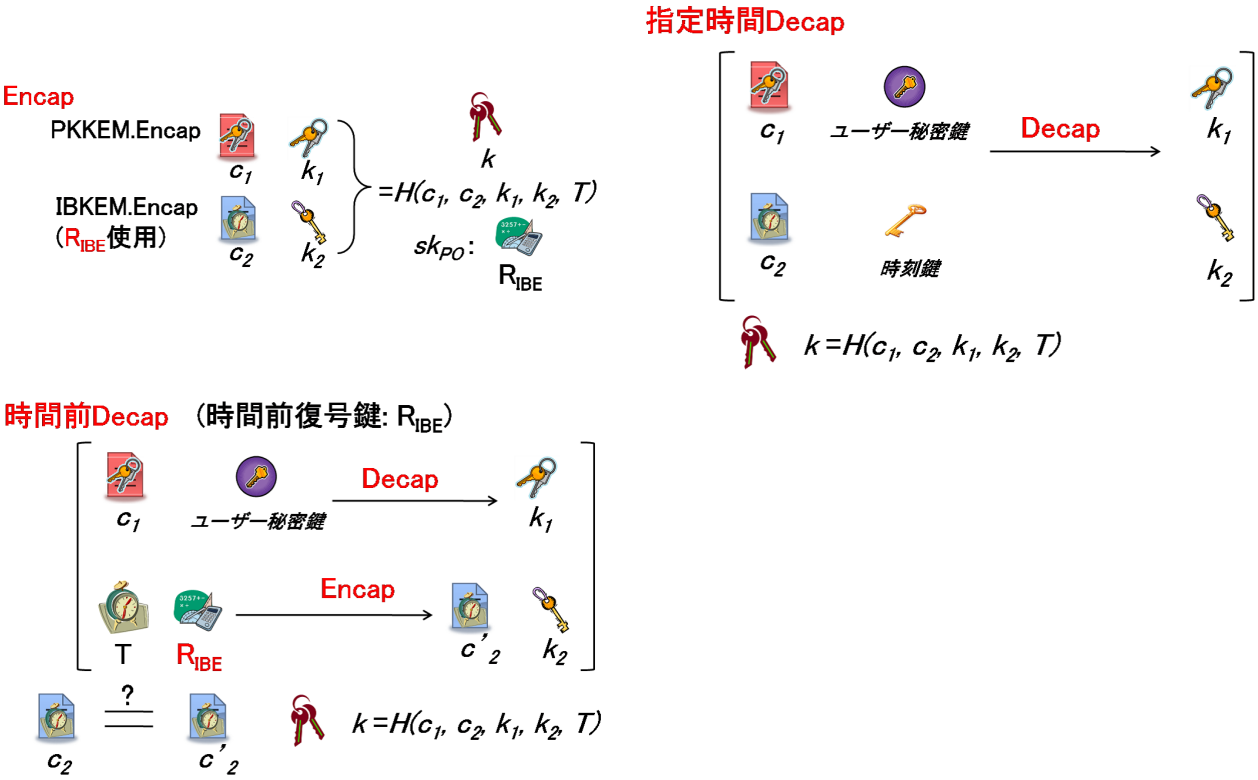

図 5.6: 提案手法 3 のイメージ図

提案方式 Γ は $\mathit{IND\text{-}TRPC\text{-}CCA}^{KEM}_{TS}$ の安全性を満たす.

証明　$\mathcal{A}$ を Γ の IND-TRPC-CCA$^{KEM}_{TS}$ ゲームを破る攻撃者とし，$\mathcal{A}$ を用いて構成要素である Π の OW-CCA 安全性を破るシミュレータ $\mathcal{S}$ を構成する．$\mathcal{S}$ は $\mathcal{A}$ に対して以下のように IND-TRPC-CCA$^{KEM}_{TS}$ ゲームのシミュレートを行いつつ利用して自身の OW-CCA チャレンジャー $\mathcal{CH}$ との間で以下のような OW-CCA ゲームを行う.

Setup. $\mathcal{S}$ は $\mathcal{CH}$ から upk を受け取る．また，IBKEM.Setup を実行し $(\mathsf{msk}, \mathsf{prm})$ を得る．$\mathcal{S}$ は $(\mathsf{msk}, \mathsf{prm}, upk)$ を $\mathcal{A}$ に渡す．また，$\mathcal{S}$ は自身の空のハッシュリストを作成する．これは，ハッシュクエリとその結果を組として記録するリストである.

Phase 1. $\mathcal{S}$ は $\mathcal{A}$ の発行する時間前復号クエリ (CT, sk_{PO})，および指定時間復号クエリ (CT, T) に対して，以下のように応答する．前者については $\mathcal{S}$ は $\mathsf{TRKEM.Decap}_{\mathsf{PO}}$ に従い CT を復号し，k を返す．後者については，$\mathcal{S}$ はまず $\mathsf{IBKEM.Ext}(\mathsf{prm}, \mathsf{msk}, T)$ を計算し d_T を得る．そして，それを用いて $\mathsf{TRKEM.Decap}_{\mathsf{TR}}$ に従って CT を復号し，k を返す．どちらの復号の場合でも，$\mathcal{S}$ は PKKEM.Decap を自分自身で行う代わりに，$\mathcal{S}$ は自らの復号クエリとして c_1 を $\mathcal{CH}$ に発行し，$\mathcal{CH}$ から返ってきた値 k_1 を用いる．また，$k \leftarrow H(c_1, c_2, k_1, k_2, T)$ を行うかわりに，(c_1, c_2, k_1, k_2, T) クエリが自身のハッシュリストに存在するかを調べ，存在した場合はその結果を k として，存在しなかった場合は，そのクエ

リに対する結果としてランダムな値を作成し，リストに加えた後にその値を k として $\mathcal{A}$ に返す．同様に，$\mathcal{A}$ はハッシュクエリ $x = (c_1, c_2, k_1, k_2, T)$ を発行することもできる．これに対して $\mathcal{S}$ は，ハッシュクエリ x が自身のハッシュクエリに存在した場合には，対応する結果として記録された値を，存在しなかった場合には，そのクエリに対する結果としてランダムな値を作成し，リストに加えた後にその値を $\mathcal{A}$ に返す．

Challenge. $\mathcal{A}$ が T^* を出力したとき，$\mathcal{S}$ は以下のようにしてチャレンジ暗号文 CT^* を作成し，$\mathcal{A}$ に返す．$\mathcal{S}$ は自身のチャレンジとして c_1^* を $\mathcal{CH}$ から受け取る．また，$R_{IBE}^* \leftarrow \mathcal{R}_{IBE}$ を一様ランダムに選び，$(c_2^*, k_2^*) \leftarrow \mathsf{IBKEM.Encap}(\mathsf{prm}, T^*; R_{IBE}^*)$ を計算する．その後，$k^* \leftarrow \mathcal{K}$ を一様ランダムに選ぶ．最後に $CT^* = (c_1^*, c_2^*, T^*), k^*$，および $sk_{PO}^* = R_{IBE}^*$ を $\mathcal{A}$ に渡す．

Phase 2. $\mathcal{S}$ は $\mathcal{A}$ の発行する時間前復号クエリ $(CT = (c_1, c_2, T), sk_{PO} = R'_{IBE})$ に対して以下のような応答を行う．ただし，ここで全て上から順に行うこととする．

(1) $c_1 \neq c_1^*$ **のとき:** Phase 1 と同様に $\mathsf{TRKEM.Decap_{PO}}$ を行い復号結果 k を返す．

(2) $c_1 = c_1^* \wedge (c_2, T) \neq (c_2^*, T^*)$ **のとき:**
$(c'_2, k_2) \leftarrow \mathsf{IBKEM.Encap}(\mathsf{prm}, T; R'_{IBE})$ を計算した後，$c_2 = c'_2$ かどうかをチェックする．$c_2 = c'_2$ のとき，$\mathcal{A}$ からのクエリ $(c_1^*, c_2, -, k_2, T)$ と自身のハッシュリストとを参照し，ハッシュリストに存在するクエリであれば，対応する出力結果を k として，そうでなければ，ランダムな値を選び，そのクエリに対する結果としてリストに加えた後に k として $\mathcal{A}$ に返す．

$c_2 \neq c'_2$ のとき $k \leftarrow \perp$ を $\mathcal{A}$ に返す．

(3) それ以外のとき: $(c_1, c_2, T) = (c_1^*, c_2^*, T^*) \wedge R'_{IBE} \neq R_{IBE}^*$ であり，それ以外の可能性はない．この時，$(c_2^*, k_2^*) = \mathsf{IBKEM.Encap}(\mathsf{prm}, T^*; R'_{IBE})$ である．ここで，このイベントを Rand とし，$\Pr[\mathsf{Rand}] = p_{rand}$ の確率で Rand を起こす IND-TRPC-CCA$_{TS}$ 攻撃者 $\mathcal{A}$ を用いて，構成要素である IB-KEM 方式 Π' の乱数に関するターゲット衝突困難性を破る攻撃者 $\mathcal{H}$ を以下のように構成する．ただし，p_{rand} は無視できない値であるとする．

Setup. $\mathcal{H}$ は $\mathcal{CH}$ から $\mathsf{msk}, \mathsf{prm}, R_{IBE}^*$ を受け取る．また，PKKEM.KG を実行し，$\mathcal{H}$ は (usk, upk) を得る．$\mathcal{H}$ は $(\mathsf{msk}, \mathsf{prm}, upk)$ を $\mathcal{A}$ に渡す．

Collision. $\mathcal{H}$ は $\mathcal{A}$ の 2 種類の復号クエリに対して，msk と usk を持っているため完全に応答できる．$\mathcal{A}$ が T^* をチャレンジとして出力した場合，$(c_1^*, k_1^*) \leftarrow \mathsf{PKKEM.Encap}(usk)$，$(c_2^*, k_2^*) \leftarrow \mathsf{IBKEM.Encap}(\mathsf{prm_I}, T^*; R_{IBE}^*)$ を計算する．また，$b_H \leftarrow \{0_0, 0_1\}$ をランダムに選び，$k_{0_0}^* = H(c_1^*, c_2^*, k_1^*, k_2^*, T^*)$，$k_{0_1}^* \leftarrow \mathcal{K}$ を計算する．$\mathcal{H}$ はチャレンジ暗号文として $CT^* = (c_1^*, c_2^*, T^*), k_{b_H}^*$，およびそれに対応する時間前復号鍵 $sk_{PO}^* = R_{IBE}^*$ を $\mathcal{A}$ に渡す．$\mathcal{A}$ が自身のチャレンジを推測した後，$\mathcal{H}$ は $\mathcal{A}$ の時間前復号クエリの中から Rand を起こすような

ものを探し，(T^*, R'_{IBE}) を出力する．そのようなクエリがなければ停止する．

上記から明らかなように，$\mathcal{H}$ は，$\mathcal{A}$ が Rand のイベントを起こす時間前復号クエリを発行するときには，必ず乱数に関するターゲット衝突困難性を破る．つまり，

$$\mathsf{Adv}^{\mathrm{Rand}}_{\Pi',\mathcal{H}} = \Pr[\mathsf{Rand}] = p_{rand}$$

となるが，これは p_{rand} が無視できない値であるという仮定より，Π' が乱数のターゲット衝突困難性を持つことと矛盾する．つまり，$\Pr[\mathsf{Rand}]$ は無視できる値となる．この結果により，このイベントが起こるという事象については無視して考えることができる

同様に，$\mathcal{S}$ は $\mathcal{A}$ の発行する指定時間復号クエリ $(CT = (c_1, c_2, T), T')$ に対して以下のような応答を行う．ただし，ここで全て上から順に行うこととし，また，$T \neq T'$ の際は全て $k \leftarrow \perp$ を返すこととする．

(1) $c_1 \neq c_1^*$ **のとき:** Phase 1 と同様に $\mathsf{TRKEM.Decap_{TR}}$ を行い復号結果 k を返す．

(2) $c_1 = c_1^*$ **のとき:**

$d_T \leftarrow \mathsf{IBKEM.Ext}(\mathsf{prm}, \mathsf{msk}, T)$, $k_2 \leftarrow \mathsf{IBKEM.Decap}(d_T, c_2)$ を計算した後，$k_2 \neq \perp$ かどうかをチェックする．$k_2 \neq \perp$ のとき，$\mathcal{A}$ からの $(c_1^*, c_2, -, k_2, T)$ と自身のハッシュリストとを参照し，ハッシュリストに入っているクエリであれば，対応する出力結果を k として，そうでなければ，ランダムな値を選び，そのクエリに対する結果としてリストに加えた後に k として $\mathcal{A}$ に返す．

$k_2 = \perp$ のとき $k \leftarrow \perp$ を $\mathcal{A}$ に返す．

また，Phase 1 と同様に，$\mathcal{A}$ はハッシュクエリ $x = (c_1, c_2, k_1, k_2, T)$ を発行することができる．これに対して $\mathcal{S}$ は，ハッシュクエリ x が自身のハッシュクエリに存在した場合には，対応する結果として記録された値を，存在しなかった場合には，そのクエリに対する結果としてランダムな値を作成し，リストに加えた後にその値を $\mathcal{A}$ に返す．

Guess. $\mathcal{A}$ が自身のチャレンジを推測した後，$\mathcal{S}$ は c_1^* の復号結果の推測として $k^*_{1b_S}$ を，ハッシュリストの中からランダムに選び出力する．

さて，ここで以下のようにイベントを定義する．

Succ:$\mathcal{A}$ が上記のゲームに勝利する．
query:$\mathcal{A}$ がハッシュクエリとして $(c_1^*, c_2^*, k_1^*, k_2^*, T^*)$ を少なくとも一度は発行する．
Rand:$\mathcal{A}$ がシミュレーションを停止させるような，すなわち，$(c_1, c_2, T) = (c_1^*, c_2^*, T^*) \wedge R'_{IBE} \neq R^*_{IBE}$ となるような時間前復号クエリを少なくとも一度は発行する．

ここで，$\mathcal{A}$ が query を起こさないとき，$\mathcal{A}$ がこのゲームで得られるアドバンテージは存在しない．なぜなら，$\mathcal{A}$ にとってはそれぞれのハッシュクエリに対する結果は完

全ランダムに見えているからである．従って，$\Pr[\mathsf{Succ}|\overline{\mathsf{query}}] = \frac{1}{2}$ となる．よって $\mathcal{A}$ のアドバンテージは以下のように計算できる．

$$\begin{aligned}
\mathsf{Adv}^{\text{IND-TRPC-CCA}_{\text{TS}}}_{\Gamma,\mathcal{A}} &= |\Pr[\mathsf{Succ}|\mathsf{query}] \cdot \Pr[\mathsf{query}] + \Pr[\mathsf{Succ}|\overline{\mathsf{query}}] \Pr[\overline{\mathsf{query}}] - \frac{1}{2}| \\
&= |\Pr[\mathsf{Succ}|\mathsf{query}] \cdot \Pr[\mathsf{query}] + \frac{1}{2}\Pr[\overline{\mathsf{query}}] - \frac{1}{2}| \\
&= |\Pr[\mathsf{Succ}|\mathsf{query}] \cdot \Pr[\mathsf{query}] - \frac{1}{2}\Pr[\mathsf{query}]| \\
&\leq \frac{1}{2}\Pr[\mathsf{query}]
\end{aligned}$$

さて，ここで $\Pr[\mathsf{query}] = p_q$ を無視できない値とする．また，$\mathcal{A}$ の発行したハッシュクエリの総数を q_H とすると，$\mathcal{A}$ は多項式時間アルゴリズムであるため，q_H は高々多項式である．

query のイベントが起こっているとき，ハッシュリストの中に少なくとも一つは，$(c_1^*, c_2^*, k_1^*, k_2^*, T^*)$ となるクエリが存在している．明らかなように，このとき $\mathcal{S}$ は自身の推測である k_1^* を当てることが可能である．つまり，$\mathcal{S}$ のアドバンテージは以下のようになる．

$$\mathsf{Adv}^{\text{OW}-\text{CCA}}_{\Pi,\mathcal{S}} \geq \frac{1}{q_H}\Pr[\mathsf{query}] = \frac{1}{q_H} \cdot p_q$$

ここで，仮定より p_q は無視できない値であるが，これは Π が OW-CCA 安全であることと矛盾する．あ

つまり，

$$\mathsf{Adv}^{\text{IND-TRPC-CCA}_{\text{TS}}}_{\Gamma,\mathcal{A}} \leq \frac{1}{2}\Pr[\mathsf{query}] = \frac{1}{2}p_q$$

となり，これは無視できる値となる．以上より，定理 5.7 は証明された．　□

IND-TRPC-CPA$^{KEM}_{IS}$

定理 5.8. 構成要素である *IB-KEM* 方式 Π' が *OW-ID-CPA* 安全性，および H がランダムオラクルとみなせるとき，提案方式 Γ は *IND-TRPC-CPA*$^{KEM}_{IS}$ の安全性を満たす．

証明　$\mathcal{A}$ を Γ の IND-TRPC-CPA$^{KEM}_{IS}$ ゲームを破る攻撃者とし，$\mathcal{A}$ を用いて構成要素である Π' の OW-ID-CPA 安全性を破るシミュレータ $\mathcal{S}$ を構成する．$\mathcal{S}$ は $\mathcal{A}$ に対して以下のように IND-TRPC-CPA$^{KEM}_{IS}$ ゲームのシミュレートを行いつつ利用して自身の OW-ID-CPA チャレンジャー $\mathcal{CH}$ は OW-ID-CPA ゲームを行う．

このときのゲームは IND-TRPC-CPA$^{KEM}_{TS}$ のときとほぼ同じとなる．異なる点は，$\mathcal{S}$ は msk でなく usk を持っている点，$\mathcal{A}$ は 2 種類の復号クエリではなく時刻鍵導出クエリ T を発行する点，そのため，Phase 2 において $\mathcal{A}$ の発行する時刻鍵導出クエリに対しては $\mathcal{S}$ 自身のクエリとして $\mathcal{CH}$ に発行することで全て応答することができるため，完全にシミュレートすることができる．

以上により定理 5.8 は証明された．　□

Binding

定理 5.9. 提案方式である Γ は計算時間に縛られないあらゆる攻撃者に対して *Binding* の安全性を満たす．

証明　$CT = (c_1, c_2, T), sk_{PO} = R_{IBE}, T'$ を Binding のゲームである攻撃者 $\mathcal{A}$ が出力したとする．このときの $\mathsf{Adv}^{\mathrm{BINDING}}_{\Gamma,\mathcal{A}}$ を計算すると以下のようになる．ただし，ここで $d_T \leftarrow \mathsf{IBKEM.Ext}(\mathsf{msk}, T), s_T \leftarrow (d_T, T)$ とする．

$$
\begin{aligned}
&\mathsf{Adv}^{\mathrm{BINDING}}_{\Gamma,\mathcal{A}} \\
&= \Pr[\mathsf{TRE.Dec_{PO}}(\mathsf{prm_I}, usk, sk_{PO}, CT) \neq \perp \wedge \mathsf{TRE.Dec_{TR}}(\mathsf{prm_I}, usk, s_T, CT) \neq \perp \\
&\qquad \wedge \mathsf{TRE.Dec_{PO}}(\mathsf{prm}, usk, sk_{PO}, CT) \neq \mathsf{TRE.Dec_{TR}}(\mathsf{prm}, usk, s_T, CT)] \\
&= \Pr[\mathsf{PKKEM.Decap}(usk, c_1) = k_1 \\
&\qquad \wedge (c_2, k_2) = \mathsf{IBKEM.Encap}(\mathsf{prm}, T; R_{IBE}) \wedge \mathsf{IBKEM.Decap}(d_T, c_2) = k_2' \\
&\qquad \wedge H(c_1, c_2, k_1, k_2, T) \neq H(c_1, c_2, k_1, k_2', T)] \\
&= \Pr[k_2 = k_2' \wedge H(c_1, c_2, k_1, k_2, T) \neq H(c_1, c_2, k_1, k_2', T)] \\
&= \Pr[k_2 = k_2' \wedge k_2 \neq k_2']
\end{aligned}
$$

ここで，最後の等式は構成要素である IBKEM 方式の正当性のためである．明らかに，$k_2 = k_2' \wedge k_2 \neq k_2'$ が成り立つことはないので，(計算時間に縛られない) あらゆる攻撃者に対して，$\mathsf{Adv}^{\mathrm{BINDING}}_{\Gamma,\mathcal{A}}$ は 0 となる．以上より，定理 5.9 は証明された．　□

Chapter 6　議論

本章では，本研究の結果についてのいくつかの議論を行う．

6.1　一般性の議論

提案手法 2 では，構成要素として IND-stag-wCCA 安全な TB-KEM を用いている．この TB-KEM という暗号方式は，一見は一般的ではない．しかし，近年提案されている効率的な PK-KEM 方式 ([14, 33] 等) は特別なコストや構成の変更なしに TB-KEM を構成することができることがわかっている [2]．PK-KEM は一般的な暗号方式であると考えられるので，その PK-KEM から簡単に構成することのできる TB-KEM を構成要素として用いることは，一般的構成法という観点から問題ないと考えている．

6.2　効率の比較

表 6.1 に本研究で示した 3 つの一般的構成法の暗号文サイズオーバーヘッドを示す．他にも TRE-PC の構成法を提案した論文は存在するが，全て具体的な数論的問題の困難性に基づく構成法であるため，一般的構成法としての暗号文オーバーヘッドの比較は本研究であげた 3 つの構成法のみで十分であると考える．

提案手法 1 ではオーバーヘッドに平文サイズおよび署名鍵のサイズが含まれてしまっている．オーバーヘッドに平文サイズが含まれると，大きな平文を暗号化する際に非常に問題となってしまい，好ましくない．また，一般的に署名によるオーバーヘッドは大きい．具体的には，内部の One-Time 署名をスタンダードモデルで最も効率の良い署名方式の一つである Boneh と Boyen による署名方式 [11] を用いて，128bit 安全性を達成しようとする場合には，1024bit 以上のサイズ増加が考えられる．つまり，$|m| + 1024$bit 以上の暗号文サイズオーバーヘッドが存在することになる．

提案手法 2 は，BK 変換 [13] のアイデアを用いており，OneTime 署名に代わり Encapsulation 方式+MAC を用いることで，効率的な構成を可能にしている．また，Encapsulation 方式についても，Matsuda ら [39] によって，非現実的な仮定に基づくことなく効率的な構成が可能なことが証明されているため，結果としては暗号文サイズを非常に小さくすることができていると考えられる．具体的には，128bit 安全性を達成しようとする場合に，BK 変換による IBE オーバーヘッド以外のオーバーヘッドが 384bit となっている．ただし，この方式は (一般的ではあると思われるが) 特殊な仮定を想定しているため，その仮定を好まない場合には BK 変換 [13] の Encapsulation を用いることになる．こちらを用いた場合であっても，128bit 安全性を達成する場合のオーバーヘッドは 704bit となっている．なおかつ，この手法では平文サイズがオーバーヘッド

表 6.1: 暗号文サイズのオーバーヘッド

方式	暗号文サイズ OH($\|c\| - \|m\|$)
提案手法 1	$\|m\|+3\|VK\|+\|\mathsf{OH_{IBE}}\|+\|\mathsf{OH_{PKE}}\|+\|\mathsf{OH_{sig}}\|+\|T\|$
提案手法 2	$\|com\| + \|\mathsf{OH_{IBKEM}}\| + \|\mathsf{OH_{TBKEM}}\| + \|\mathsf{OH_{SKE}}\| + \|d\| + \|\mathsf{OH_{MAC}}\| + \|T\|$
提案手法 3	$\|\mathsf{OH_{IBKEM}}\| + \|\mathsf{OH_{PKKEM}}\| + \|\mathsf{OH_{SKE}}\| + \|T\|$

$|m|, |VK|, |com|, |d|, |T|$ はそれぞれ平文，One-time 署名の検証鍵，コミットメント，デコミットメント，時間のサイズを表す．また $|\mathsf{OH}|$ はそれぞれの方式のオーバーヘッドを表す．
† 提案手法 3 のオーバーヘッドは，構成した TRE-PC KEM と SKE を組み合わせて TRE-PC としたもののオーバーヘッドを考えている．
‡ 提案手法 3 はランダムオラクルモデルであり，その他は使用する構成要素の安全性モデルに従う．

に存在しない．つまり，どんな長さの平文を暗号化しても生じるオーバーヘッドが一定となる点も，大きな利点となっている．

提案手法 3 については，ランダムオラクルモデルではあるものの，暗号文サイズのオーバーヘッドに構成要素であるそれぞれの暗号方式 PK-KEM, IB-KEM のオーバーヘッドしか存在せず，非常に効率のよいものになっている．例えば，この手法を用いて Boneh-Franklin-IBE[12] と，DHIES[3]，もしくは Twin Elgamal[17] を組み合わせれば，具体的な数論の困難性に基づいた方式を含めても，現時点で最も効率のよい TRE-PC を構成できる可能性がある．現実に実装を考えた際には，ランダムオラクルモデルで充分とされる場合も多くあるため，この構成法についても有意義なものであると考えられる．

また，この結果だけから見ると，提案手法 1 の結果は価値のないものと感じるかもしれないが，提案手法 1 については，(時間前開封機能を持たない) 普通の TRE 方式の一般的構成法と実質的には全く同じ構成で TRE-PC を構成しているという点が大変興味深く，提案手法 2，3 についてはこの結果を起点として改良したものとなっているので，十分に価値のあるものであると考えている．

6.3　今後の課題

今後の課題としては，大きなものとして実装がある．暗号技術，特に，ID ベース暗号を用いた暗号技術については，標準化されたプラットフォームが存在していないため，実装をする際には完全に最初から作り始めなければならず，また，そのノウハウが他で活かされることも少ない．結果として，ID ベース暗号の理論研究が盛んに行われているのに比べて，実装を行ったものは非常に少なくなってしまっている．

そこで，ID ベース暗号を用いた実装をより簡単に行えるように，プラットフォームの標準化についての取り組みを行い，公の場所での発表も行っている (発表文献 [iv][v][vi]).

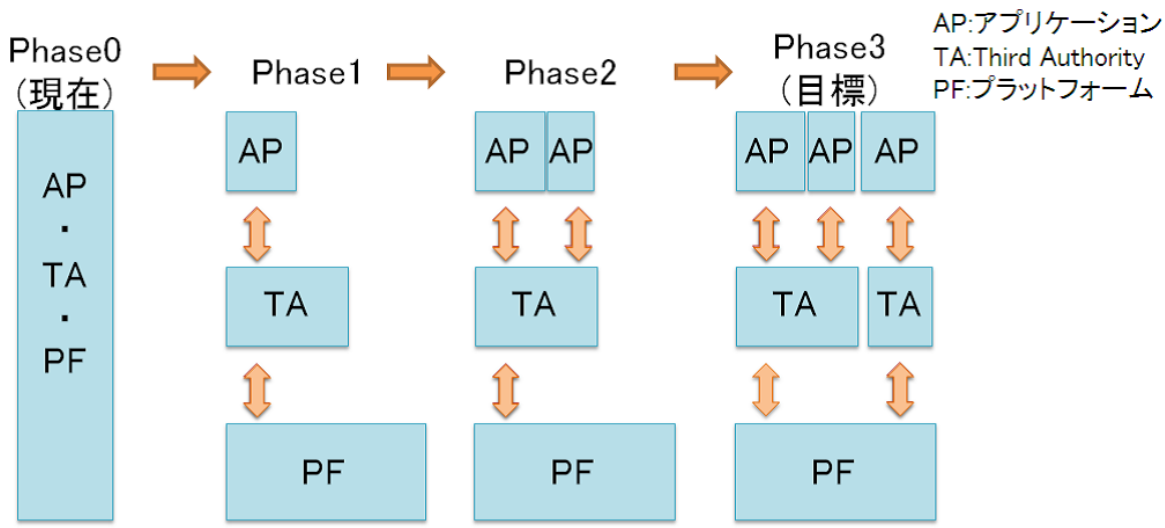

図 6.1: プラットフォームのイメージ図

この取り組みの概要としては，ID ベース暗号には TA(Third Authority) と呼ばれる機関 (TRE でいう TS のようなエンティティ) が存在するのだが，アプリケーションを実装する際に TA から作る必要のあるものではなく，今回標準化するプラットフォームに則って実装されたアプリケーションであれば，一つの TA を複数のアプリケーションで共用できるようにする，というものである (図 6.1)．このようにすることで，ID ベース暗号を利用したアプリケーションを実装するコストから，TA を構成する，というコストを省くことができるため，より容易に実装を行うことができるようになると考えられる．

また，そのプラットフォーム策定の一環として，実際に TRE の実装も並行して行っている．この結果として構成した TA を公開し，様々なところで使用してもらうことで，さらに議論を重ね，よりよいプラットフォームを策定することが今後の課題である．

Chapter 7　結論

本研究では，時間前開封機能付き時限式暗号 (TRE-PC) について着目し，今まで存在していなかった，一般的構成法を目指した．

そして，3 種類の一般的構成法を提案し，その安全性を証明した．提案方式の一つ目は，公開鍵暗号，ID ベース暗号，および電子署名を構成要素とする構成法である．興味深いことに，この構成法は，既存の (時間前開封機能のない) 時限式暗号の一般的構成法と本質的には同じ構成をしているため，もし既に一般的構成法で作られた時限式暗号があれば，それを容易に時間前開封機能の付いた時限式暗号にすることができる可能性を持っている．

提案方式の二つ目は，タグベース鍵カプセル化メカニズム，ID ベース鍵カプセル化メカニズム，Encapsulation，共通鍵暗号，および，メッセージ認証子を構成要素とする構成法である．一つ目の構成法では，電子署名を構成要素としているが，電子署名は暗号文サイズが大きくなってしまうため，電子署名の代わりに Encapsulation 方式，およびメッセージ認証子を用いることで，暗号文サイズを小さくすることに成功している．

提案方式の三つめは，鍵カプセル化メカニズム，および，ID ベース鍵カプセル化メカニズムを構成要素とする構成法である．この方式は，ランダムオラクルモデルでの安全性の証明を行ったものであるため，上記の二つの構成法に比べると安全性は弱いものとなっているが，ランダムオラクルモデルである，ということに対する実用上の問題点は現在のところわかっておらず，また，一般的には効率が悪くなってしまうと考えられている一般的構成法にも関わらず，この方式を用いて既存の暗号技術から TRE-PC を構成すると，既存の具体的な数論の困難性に基づいて安全性の証明がなされている TRE-PC の構成法と比べても，暗号文サイズの観点で効率のよい構成ができる可能性を持っているものとなっている．

謝辞

本研究にあたり，日頃から常にご指導を頂きました東京大学生産技術研究所 松浦幹太准教授に心から感謝致します．松浦先生には，研究の内容や進め方，考え方に関するご指導だけでなく，研究に対する取り組み方や着眼点，また，私が働くようになった時に必要になると思われるようなことまで教えていただき，また，学会参加など様々な活動の機会を与えていただいたことで，修士 2 年間を非常に有意義なものにすることができました．

また，松浦研の定期ミーティングでの発表時に，研究内容に関する議論や，助言をくださった，NHK 放送技術研究所の小川一人さん，警察庁情報技術解析課の岡田智明さん，情報処理推進機構の野島良さんを始めとする今まで修士 2 年間の間にお世話になった松浦研ミーティングの参加者の皆様に感謝致します．

そして，私たちの研究活動が円滑に進むように日頃から尽力してくださっている教授室秘書であった仲野さん，また現教授室秘書の橋詰さんに心より感謝を致します．学会出張などの際に至らないところが多々あったと思いますが，見えないところでフォローをしていただきました．

また，松浦研の技術職員である細井琢朗さん，松浦研メンバーである楊鵬さん，Jacob Schuldt さん，松田隆宏さん，北田亘さん，渡邉悠さん，施屹君，千葉大輝君，Bongkot Jenjarrussakul さん，付紹静さんには，日頃から研究室内での議論や，松浦研定期ミーティングにおいて，活発に議論をしたり，適切な助言をいただきました．特に松田さんには，研究の内容に関することから，論文の書き方，発表資料の細部に至るまで，自身の研究もお忙しい中，様々な面で助言をいただきました．改めて，深く感謝致します．皆様のおかげで松浦研究室での 2 年間の生活は，楽しく，また，素晴らしいものとなりました．

友人達にも感謝します．ある時は就職活動などの相談に，ある時は気分転換に付き合ってもらい，研究活動とそれ以外の活動のメリハリをつけ，より気分良く研究をする大きな手助けとなりました．

最後に，常日頃から私を支えてくれた家族に心から感謝します．

参考文献

[1] *PKCS #1 v2.1: RSA Cryptography Standard, RSA Laboratories, June 14, 2002.*

[2] Takahiro Matsuda 0002, Kanta Matsuura, and Jacob C. N. Schuldt. Efficient constructions of signcryption schemes and signcryption composability. In *INDOCRYPT*, volume 5922 of *LNCS*, pages 321–342. Springer, 2009.

[3] Michel Abdalla, Mihir Bellare, and Phillip Rogaway. The oracle diffie-hellman assumptions and an analysis of dhies. In *CT-RSA*, volume 2020 of *LNCS*, pages 143–158. Springer, 2001.

[4] 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 *EUROCRYPT*, pages 128–146, 2005.

[5] Sattam S. Al-Riyami and Kenneth G. Paterson. Certificateless public key cryptography. In *ASIACRYPT*, volume 2894 of *LNCS*, pages 452–473. Springer, 2003.

[6] Nuttapong Attrapadung, Yang Cui, David Galindo, Goichiro Hanaoka, Ichiro Hasuo, Hideki Imai, Kanta Matsuura, Peng Yang, and Rui Zhang. Relations among notions of security for identity based encryption schemes. In *LATIN*, pages 130–141, 2006.

[7] Mihir Bellare, Alexandra Boldyreva, and Adriana Palacio. An uninstantiable random-oracle-model scheme for a hybrid-encryption problem. In *EUROCRYPT*, volume 3027 of *Lecture Notes in Computer Science*, pages 171–188. Springer, 2004.

[8] Mihir Bellare and Phillip Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In *ACM Conference on Computer and Communications Security*, pages 62–73, 1993.

[9] Mihir Bellare and Phillip Rogaway. The exact security of digital signatures - how to sign with rsa and rabin. In *EUROCRYPT*, pages 399–416, 1996.

[10] Mihir Bellare and Phillip Rogaway. Collision-resistant hashing: Towards making uowhfs practical. In *CRYPTO*, volume 1294 of *Lecture Notes in Computer Science*, pages 470–484. Springer, 1997.

[11] Dan Boneh and Xavier Boyen. Secure identity based encryption without random oracles. In *CRYPTO*, volume 3152 of *LNCS*, pages 443–459. Springer, 2004.

[12] Dan Boneh and Matthew K. Franklin. Identity-based encryption from the weil pairing. In *CRYPTO*, volume 2139 of *LNCS*, pages 213–229. Springer, 2001.

[13] Dan Boneh and Jonathan Katz. Improved efficiency for cca-secure cryptosystems built using identity-based encryption. In *CT-RSA*, volume 3376 of *LNCS*, pages 87–103. Springer, 2005.

[14] Xavier Boyen, Qixiang Mei, and Brent Waters. Direct chosen ciphertext security from identity-based techniques. In *ACM Conference on Computer and Communications Security*, pages 320–329, 2005.

[15] Ran Canetti, Oded Goldreich, and Shai Halevi. The random oracle methodology, revisited (preliminary version). In *STOC*, pages 209–218, 1998.

[16] Ran Canetti, Shai Halevi, and Jonathan Katz. Chosen-ciphertext security from identity-based encryption. In *EUROCRYPT*, volume 3027 of *LNCS*, pages 207–222. Springer, 2004.

[17] David Cash, Eike Kiltz, and Victor Shoup. The twin diffie-hellman problem and applications. In *EUROCRYPT*, volume 4965 of *LNCS*, pages 127–145. Springer, 2008.

[18] Julien Cathalo, Benoît Libert, and Jean-Jacques Quisquater. Efficient and non-interactive timed-release encryption. In *ICICS*, volume 3783 of *LNCS*, pages 291–303. Springer, 2005.

[19] Konstantinos Chalkias, Dimitrios Hristu-Varsakelis, and George Stephanides. Improved anonymous timed-release encryption. In *ESORICS*, volume 4734 of *LNCS*, pages 311–326. Springer, 2007.

[20] Konstantinos Chalkias and George Stephanides. Timed release cryptography from bilinear pairings using hash chains. In *Communications and Multimedia Security*, volume 4237 of *LNCS*, pages 130–140. Springer, 2006.

[21] Aldar C.-F. Chan and Ian F. Blake. Scalable, server-passive, user-anonymous timed release cryptography. In *ICDCS*, pages 504–513. IEEE Computer Society, 2005.

[22] Jung Hee Cheon, Nicholas Hopper, Yongdae Kim, and Ivan Osipkov. Timed-release and key-insulated public key encryption. In *Financial Cryptography*, volume 4107 of *LNCS*, pages 191–205. Springer, 2006.

[23] Jung Hee Cheon, Nicholas Hopper, Yongdae Kim, and Ivan Osipkov. Provably secure timed-release public key encryption. volume 11, 2008.

[24] Sherman S. M. Chow, Volker Roth, and Eleanor G. Rieffel. General certificateless encryption and timed-release encryption. In *SCN*, volume 5229 of *LNCS*, pages 126–143. Springer, 2008.

[25] Sherman S. M. Chow and Siu-Ming Yiu. Timed-release encryption revisited. In *ProvSec*, volume 5324 of *LNCS*, pages 38–51. Springer, 2008.

[26] T. Coffey and P. Saidha. Logic for verifying public-key cryptographic protocols. In *Computers and Digital Techniques, IEE Proceedings-*, volume 144, pages 28–32, 1997.

[27] Giovanni Di Crescenzo, Rafail Ostrovsky, and Sivaramakrishnan Rajagopalan. Conditional oblivious transfer and timed-release encryption. In *EUROCRYPT*, pages 74–89, 1999.

[28] Alexander W. Dent and Qiang Tang. Revisiting the security model for timed-release encryption with pre-open capability. In *ISC*, volume 4779 of *LNCS*, pages 158–174. Springer, 2007.

[29] W. Diffie and M. Hellman. New directions in cryptography. volume 22, pages 644–654, 1976.

[30] Amos Fiat and Adi Shamir. How to prove yourself: Practical solutions to identification and signature problems. In *CRYPTO*, volume 263 of *Lecture Notes in Computer Science*, pages 186–194. Springer, 1986.

[31] Craig Gentry. Practical identity-based encryption without random oracles. In *EUROCRYPT*, volume 4004 of *LNCS*, pages 445–464. Springer, 2006.

[32] Shafi Goldwasser and Yael Tauman Kalai. On the (in)security of the fiat-shamir paradigm. In *FOCS*, pages 102–. IEEE Computer Society, 2003.

[33] Goichiro Hanaoka and Kaoru Kurosawa. Efficient chosen ciphertext secure public key encryption under the computational diffie-hellman assumption. In *ASIACRYPT*, volume 5350 of *LNCS*, pages 308–325. Springer, 2008.

[34] Dimitrios Hristu-Varsakelis, Konstantinos Chalkias, and George Stephanides. Low-cost anonymous timed-release encryption. In *IAS*, pages 77–82. IEEE Computer Society, 2007.

[35] Yong Ho Hwang, Dae Hyun Yum, and Pil Joong Lee. Timed-release encryption with pre-open capability and its application to certified e-mail system. In *ISC*, volume 3650 of *LNCS*, pages 344–358. Springer, 2005.

[36] Eike Kiltz. Chosen-ciphertext security from tag-based encryption. In *TCC*, volume 3876 of *LNCS*, pages 581–600. Springer, 2006.

[37] Michiharu Kudo and Anish Mathuria. An extended logic for analyzing timed-release public-key protocols. In *ICICS*, volume 1726 of *LNCS*, pages 183–198. Springer, 1999.

[38] Wenbo Mao. Timed-release cryptography. In *Selected Areas in Cryptography*, pages 342–358. Springer-Verlag London, UK, 2001.

[39] Takahiro Matsuda, Goichiro Hanaoka, Kanta Matsuura, and Hideki Imai. An efficient encapsulation scheme from near collision resistant pseudorandom generators and its application to ibe-to-pke transformations. In *CT-RSA*, volume 5473 of *LNCS*, pages 16–31. Springer, 2009.

[40] T. May. Timed-release crypto. *Unpublished manuscript*, 1993.

[41] Deholo Nali, Carlisle M. Adams, and Ali Miri. Time-based release of confidential information in hierarchical settings. In *ISC*, volume 3650 of *LNCS*, pages 29–43. Springer, 2005.

[42] Moni Naor and Moti Yung. Universal one-way hash functions and their cryptographic applications. In *STOC*, pages 33–43. ACM, 1989.

[43] Jesper Buus Nielsen. Separating random oracle proofs from complexity theoretic proofs: The non-committing encryption case. In *CRYPTO*, volume 2442 of *LNCS*, pages 111–126. Springer, 2002.

[44] I. Osipkov, Y. Kim, and JH Cheon. Timed-Release Public Key Based Authenticated Encryption. Technical report, Cryptology ePrint Archive, 2004.

[45] RL Rivest, A. Shamir, and DA Wagner. Time-lock Puzzles and Timed-release Crypto. *MIT LCS Tech. Report MIT/LCS/TR-684*, 1996.

[46] Ronald L. Rivest, Adi Shamir, and Leonard M. Adleman. A method for obtaining digital signatures and public-key cryptosystems. volume 21, pages 120–126, 1978.

[47] Adi Shamir. Identity-based cryptosystems and signature schemes. In *CRYPTO*, pages 47–53, 1984.

[48] Victor Shoup. Using hash functions as a hedge against chosen ciphertext attack. In *EUROCRYPT*, pages 275–288, 2000.

[49] Brent Waters. Efficient identity-based encryption without random oracles. In *EUROCRYPT*, volume 3494 of *LNCS*, pages 114–127. Springer, 2005.

[50] Maki Yoshida, Shigeo Mitsunari, and Toru Fujiwara. A timed-release key management scheme for backward recovery. In *ICISC*, volume 3935 of *LNCS*, pages 3–14. Springer, 2005.

[51] 齊藤泰一 岡本義明. 復元信号を用いた公開鍵型 timed-release 暗号. In **コンピュータセキュリティシンポジウム** *(CSS 2008)*, 2008.

[52] 藤原融 吉田真紀, 光成滋生. タイムカプセル暗号. In **電子情報通信学会技術研究報告**. *ISEC*, volume 104, pages pp. 1–5, 2004.

[53] 今井秀樹 古川倫章, 崔洋. 弱い仮定に基づく TRE-PC. In **暗号と情報セキュリティシンポジウム** *(SCIS 2009)* **予稿集** *CDROM, 3F2-2*, 2009.

[54] 小川貴英 今井秀樹 繁富利恵, 大塚玲. 期限付き匿名貸し出しプログラムにおける一考察. In **情報処理学会研究報告**. *CSEC*, volume 2002, pages pp. 153–160, 2002.

発表文献

査読付き国際会議投稿論文

i Yasumasa Nakai, Takahiro Matsuda, Wataru Kitada, Kanta Matsuura. “A Generic Construction of Time-Release Encryption with Pre-open Capability”, In *IWSEC, volume 5824 of LNCS, pages 53-70,. Springer, 2009.*

査読無し国内会議投稿論文

ii 中井泰雅, 松田隆宏, 北田亘, 松浦幹太. “時間前開封機能付き時限式暗号の一般的構成法”, 暗号と情報セキュリティシンポジウム (SCIS 2009) 予稿集 CDROM, 3F2-2, 2009.

iii 中井泰雅, 松田隆宏, 松浦幹太. “時間前復号機能付き時限式暗号の効率的な一般的構成法”, 暗号と情報セキュリティシンポジウム (SCIS 2010) 予稿集 CDROM, 2C3-1, 2010.

iv 小田 哲, 永井 彰, 山本 剛, 小林 鉄太郎, 冨士 仁, 中井 泰雅, 松田 隆宏, 松浦 幹太 “汎用 IBE 向けシステムの構成法とその実装”, 暗号と情報セキュリティシンポジウム (SCIS 2010) 予稿集 CDROM, 4C2-5, 2010.

研究会発表

v 小林鉄太郎, 中井泰雅. “タイムリリース暗号システムの設計と実装”, 第 10 回 ペアリングフォーラム, 2009.

vi 中井泰雅　他. “タイトル未定”, 第 11 回 ペアリングフォーラム, 2010(発表予定).