RSO-CCA 安全性を満たす ID ベース暗号

原　啓祐[1,2,a)]　田中　圭介[1]

概要：ID ベース暗号の Receiver Selective Opening (RSO) 安全性は、一人の送信者と多数の受信者がいる状況において、攻撃者が一部の受信者のユーザ秘密鍵と受信した平文を取得できる際に、その他の受信者の暗号文の秘匿性を保証する。Kitagawa と Tanaka (PKC 2018) は、RSO 安全性を満たす ID ベース暗号を初めて提案した。彼らの方式は、選択平文攻撃 (CPA) に対してのみ安全 (RSO-CPA 安全) であり、選択暗号文攻撃 (CCA) に対しては安全でない。本論文では、RSO-CCA 安全性を満たす ID ベース暗号の定義とその構成方法を与える。

キーワード：RSO 安全性、選択暗号文攻撃、ID ベース暗号

RSO-CCA Secure Identity-based Encryption

KEISUKE HARA[1,2,a)]　KEISUKE TANAKA[1]

Abstract: In the situation where there are one sender and multiple receivers, a receiver selective opening (RSO) attack for an identity-based encryption (IBE) scheme considers adversaries that can corrupt some of the receivers and get their secret keys and plaintexts. Security against RSO attacks for an IBE scheme ensures confidentiality of ciphertexts of uncorrupted receivers. Kitagawa and Tanaka (PKC 2018) proposed the first RSO secure IBE scheme against chosen plaintext attacks (CPA). To the best of our knowledge, an IBE scheme with RSO security against chosen ciphertext attacks (RSO-CCA security) has not been explored so far. In this paper, we formalize RSO-CCA security for IBE and propose the first RSO-CCA secure IBE scheme.

Keywords: RSO Security, Chosen Ciphertext Attack, Identity-based Encryption

1. 導入

1.1 背景と動機

Bellare, Hofheinz, Yilek [3] は、暗号化方式を多数のユーザ間で用いる状況において、攻撃者が一部のユーザの秘密情報を取得できるとしても、その他のユーザの暗号文の秘匿性が保たれるという Selective Opening (SO) 安全性を初めて定式化した。実社会での暗号化方式の利用を考えた際、多数のユーザ間で秘匿通信を行うことが想定されるため、SO 安全性は実用的に重要な安全性である。

SO 安全性には、次の二種類が考えられてきた。一つ目は、多数の送信者と一人の受信者がいる状況において、一部の送信者が暗号化時に用いた乱数と平文を攻撃者が取得できるとしても、その他の送信者の暗号文の秘匿性が保証される Sender Selective Opening (SSO) 安全性 [3, 5] である。二つ目は、一人の送信者と多数の受信者がいる状況において、一部の受信者の秘密鍵と平文を攻撃者が取得できるとしても、その他の受信者の暗号文の秘匿性が保証される Receiver Selective Opening (RSO) 安全性 [2, 9] である。SSO 安全性と RSO 安全性のそれぞれについて、選択平文攻撃 (CPA) と選択暗号文攻撃 (CCA) の両攻撃に対する定義を考えることができる。一般的に暗号方式では、能動的な攻撃者を考慮するためには CCA 安全性が必要である。よって、CCA 安全性は CPA 安全性よりも望ましい安全性であると言える。

ID ベース暗号は、Shamir [19] によって提案された公開鍵暗号を発展させた暗号技術である。ID ベース暗号では、受信者の ID を公開鍵として送信者が暗号文を生成可能である。受信者は自身の ID に対応する秘密鍵を、マスター秘密鍵を持った信頼できる第三者機関から発行してもらい、その秘密鍵を用いて自身の ID の元で生成された暗号文を復号可能である。ID ベース暗号を用いる最大のメリットとして、公開鍵暗号を利用する際の問題の一つである公開鍵証明書配布の問題を解決できることが挙げられる。

従来より ID ベース暗号には標準的な安全性として、適応的 ID 攻撃に対して安全 [6] (IND-ID-CPA 安全) であることが求められてきた。攻撃者が攻撃対象の ID id^* を適応的に選択し、かつ、id^* 以外のいかなる ID に対応する秘密鍵を取得したとしても、id^* の下で暗号化された平文の秘匿性を保証できるときに、ID ベース暗号は IND-ID-CPA 安全であるという。

しかしながら、ID ベース暗号はその暗号技術としての性質上、自然な利用状況として複数の攻撃対象のユーザが存

[1] 東京工業大学 情報理工学院、〒 152-8552 東京都目黒区大岡山 2-12-1-W8-55.
[2] 国立研究開発法人産業技術総合研究所 (AIST)。
[a)] hara.k.am@m.titech.ac.jp

本研究の一部は Input Output Hong Kong、野村総合研究所、NTT セキュアプラットフォーム研究所、三菱電機、アイ・システム、JST CREST JPMJCR14D6、JST OPERA、JSPS 科研費 16H01705, 16J10322, 17H01695 の助成を受けています。
本研究の一部は JST CREST JPMJCR1688 の支援を受けて行われた。

在する状況が想定され、それらの一部のユーザの秘密鍵が攻撃者に漏洩することも考慮するべきである。このような動機から、Kitagawa と Tanaka [13] は、ID ベース暗号に対して RSO-CPA 安全性を導入した。そして彼らは、RSO-CPA 安全性を満たす ID ベース暗号方式が、IND-ID-CPA 安全性を満たす ID ベース暗号方式のみから構成可能であることを示した。しかしながら、彼らの方式は CCA 安全性を満たすことは出来ず、現在まで RSO-CCA 安全性を満たす ID ベース暗号方式は知られていない。

前述の通り、CCA 安全性を満たす暗号方式は能動的な攻撃者に対しても安全性を保証可能であるため、ID ベース暗号の実利用を考えた際、RSO-CCA 安全性を満たすことがより望まれる。そこで、本論文では RSO-CCA 安全性を満たす ID ベース暗号方式の構成を目指す。

1.2 本論文の貢献

本論文で我々は、ID ベース暗号に対する RSO-CCA 安全性の定義を新たに与え、RSO-CCA 安全な ID ベース暗号方式の一般的構成を与える。具体的には、IND-ID-CPA 安全性を満たす ID ベース暗号方式と非対話型ゼロ知識証明システムを用いて、RSO-CCA 安全な ID ベース暗号方式が構成可能であることを示す。IND-ID-CPA 安全性を満たす ID ベース暗号方式と非対話型ゼロ知識証明システムは、暗号学的な双線型写像上の困難性仮定に基づく構成 [7,8] や格子理論の困難性仮定に基づく構成 [1,17] などが知られているため、我々の提案方式によりこれらの困難性仮定に基づいた構成が可能であることが示される。

1.3 構成の概要

我々の RSO-CCA 安全性を満たす ID ベース暗号方式は、Kitagawa と Tanaka [13] の RSO-CPA 安全性を満たす ID ベース暗号方式の構成を拡張させたものであるため、まず初めに彼らの構成を振り返る。

RSO-CPA 安全性を満たす ID ベース暗号方式の構成方法

Kitagawa と Tanaka [13] の構成は、古典的な Naor-Yung 構成 [14] を発展させた構成である。以下では、ID 空間 $\mathcal{ID} \times \{0,1\}$ と平文空間 $\{0,1\}$ をもつ ID ベース暗号方式を Π として、ID 空間 $\mathcal{ID}$ と平文空間 $\{0,1\}$ をもつ彼らの ID ベース暗号方式 Π' の構成を示す。

Π' のセットアップアルゴリズムは、Π と同じである。Π' の鍵生成アルゴリズムは、ID $id \in \mathcal{ID}$ に対するユーザ秘密鍵 SK_{id} を生成する際、まずビット $\alpha \in \{0,1\}$ を一様ランダムに選び、ID (id, α) に対するユーザ秘密鍵 $sk_{id,\alpha}$ を生成し、$SK_{id} := (\alpha, sk_{id,\alpha})$ を出力する。ID $id \in \mathcal{ID}$ の下で平文 $m \in \{0,1\}$ を暗号化する場合、Π' の暗号化アルゴリズムは、全ての $i \in \{0,1\}$ に対して ID (id, i) の下での平文 m の暗号文 c_i を生成し、$c := (c_0, c_1)$ を出力する。Π' の復号アルゴリズムは、暗号文 $c = (c_0, c_1)$ とユーザ秘密鍵 $SK_{id} = (\alpha, sk_{id,\alpha})$ を受け取り、$sk_{id,\alpha}$ を用いて c_α の復号結果を出力する。

上記の通り構成された新たな ID ベース暗号方式 Π' は、ユーザ秘密鍵に関する non-committing 性を達成している。具体的には、ID id の下での暗号文 $c = (c_0, c_1)$ を生成する際に、c_α を ID (id, α) の下での 0 の暗号文として生成し、$c_{1\oplus\alpha}$ を ID $(id, 1 \oplus \alpha)$ の下での 1 の暗号文として生成すると仮定する。(ただし、$\alpha \in \{0,1\}$ は ID id に対するユーザ秘密鍵の生成時に用意した乱数である。) このような “フェイク” の暗号文に対して、ID id に対するユーザ秘密鍵 SK_{id} を $(\alpha \oplus m, sk_{id,\alpha\oplus m})$ と用意すれば、この暗号文を任意の平文 $m \in \{0,1\}$ に開示することが可能である。秘密鍵に関する non-committing 性を持つ公開鍵暗号方式が、RSO-CPA 安全な公開鍵暗号を含意する [9] ことが知られており、彼らの構成についても同様に、ユーザ秘密鍵に関する non-committing 性により、Π' が RSO-CPA 安全性を満たすことを証明可能である。

RSO-CPA 安全性から RSO-CCA 安全性への変換

我々は、公開鍵暗号において IND-CCA 安全性を達成する代表的手法である double encryption technique [16] を上記の RSO-CPA 安全な ID ベース暗号方式に適用することで、RSO-CCA 安全性を満たす ID ベース暗号方式を構成する。簡単に言えば、公開鍵暗号における double encryption technique とは、IND-CPA 安全性を満たす公開鍵暗号方式と一回シミュレーション健全性を満たす非対話型ゼロ知識証明システムを用いて、IND-CCA 安全性を満たす公開鍵暗号方式を構成する手法である。

ここで、RSO-CPA 安全な ID ベース暗号方式に対して double encryption technique を適用する際に注意しなくてはならない点がある。それは、構成要素である非対話型ゼロ知識証明システムが、一回シミュレーション健全性とゼロ知識性を満たしているだけでは不十分であり、より強力な “unbound simulation soundness” と “multi-theorem zero knowledge” と呼ばれる二つの性質を満たしている必要がある点である。これは、公開鍵暗号方式では、鍵生成時に各ユーザの公開鍵にそのユーザだけが用いる非対話型ゼロ知識証明システムの共通参照情報を含むことが可能であったのに対し、ID ベース暗号方式では、各ユーザは個々に公開鍵を持たないために、セットアップ時に作成された一つの共通参照情報を全ユーザで共有する必要があるためである。詳細については、3.2 節を参照されたい。

本論文では、平文空間が $\{0,1\}$ であるような RSO-CCA 安全な ID ベース暗号方式のみを与えるが、平文空間を多ビットに拡張した方式も構成可能である。具体的には、上記の RSO-CPA 安全性を満たす ID ベース暗号方式の構成を並列に用いて、それら全体に非対話型ゼロ知識証明システムの証明を生成することにより構成できる。

1.4 関連研究

公開鍵暗号に対する RSO 安全性

Bellare, Dowsley, Waters, Yilek [2] は、公開鍵暗号における RSO-CPA 安全性を初めて導入し、衝突困難ハッシュ関数の存在下で、IND-CPA 安全性と RSO-CPA 安全性の間に差があることを示した。Hazay, Patra, Warinschi [9] は、秘密鍵に関する non-committing 性を持った特殊な公開鍵暗号である Receiver Non-committing Encryption (RNCE) は、RSO-CPA 安全性を満たす公開鍵暗号を含意することを示した。Jia, Lu, Li [12] は、公開鍵暗号に対する RSO-CCA 安全性を導入し、識別不可能性難読化、擬似乱数生成器、及び、穴あき擬似ランダム関数を用いて RSO-CCA 安全性を満たす公開鍵暗号を構成可能であることを示した。Huang, Lai, Chen, Au, Peng, Li [11] は、Decisional Diffie-Hellman (DDH) 仮定、または、Decisional Composite residuosity (DCR) 仮定に基づいた RSO-CCA 安全な公開鍵暗号方式を提案した。さらに、マスター秘密鍵に関する RSO 安全性を満たす ID ベース暗号方式を用いて、RSO-CCA 安全性を満たす公開鍵暗号方式を構成可能であることを示した。Hara、Kitagawa、Matsuda、Hanaoka、Tanaka [10] は、RNCE に対して CCA 安全性を導入し、CCA 安全な RNCE から RSO-CCA 安全性を満たす公開鍵暗号方式を構成可能であることを示した。そして、IND-CPA 安全な公開鍵暗号方式と非対話型ゼロ知識証明システム、もしくは、DDH 仮定に基づいて CCA 安全な RNCE が構成可能であることを示した。

ID ベース暗号に対する SSO 安全性

Bellare, Waters, Yilek [4] は、ID ベース暗号に対する SSO-CPA 安全性を導入し、decisional linear 仮定、または、(合成数位数の双線型群上の) subgroup decision 仮定に基づいて SSO-CPA 安全性を満たす ID ベース暗号方式を構成可能であることを示した。さらに、Lai, Deng, Liu, Weng, Zhao [15] は、ID ベース暗号に対する SSO-CCA 安全性を導入し、SSO-CCA 安全性を満たす初めての ID ベース暗号方式を提案した。具体的には彼らは、通常の ID ベース

暗号を発展させた “抽出可能” ID ベース暗号を新たに導入し、“One-sided Public Openability” と呼ばれる安全性を満たす抽出可能 ID ベース暗号方式、衝突困難ハッシュ関数、及び、Cross-authentication Code を用いて、SSO-CCA 安全性を満たす ID ベース暗号方式を構成可能であることを示した。これまでに提案された ID ベース暗号に対する SSO-CPA(CCA) 安全性は、適応的 ID 攻撃を捉えた安全性である。

2. 準備

本章では、本論文で用いるいくつかの記法と暗号学的要素技術を導入する。

2.1 記法

本論文では、$x \leftarrow X$ は有限集合 X から要素を一様ランダムに選ぶことを表す。また、$y \leftarrow \mathcal{A}(x;r)$ は確率的アルゴリズム $\mathcal{A}$ が入力 x に対し、乱数 r を用いて y を出力することを表し、内部乱数 r を明示的に書く必要がないときは $y \leftarrow \mathcal{A}(x)$ と略記する。そして、$y := x$ は要素 x を要素 y に代入することを表す。λ はセキュリティパラメータを表し、関数 $f(\lambda)$ が全ての定数 $c > 0$ に対して $\frac{1}{\lambda^c}$ よりも早く 0 に収束するとき、$f(\lambda)$ は無視可能であるという。ある関数 f が無視可能であることを $f(\lambda) = \mathsf{negl}(\lambda)$ によって表す。確率的多項式時間を PPT と略記する。$[n]$ は整数の集合 $\{1, \cdots, n\}$ を表し、$[a, b]$ は整数の集合 $\{a, \cdots, b\}$ を表す。$\mathbf{m} = (m_1, \cdots, m_n)$ が n 次元のベクトルであるとき、$\mathbf{m}_{\mathcal{J}}$ は $\mathcal{J} \subseteq [n]$ に対する $\mathbf{m}$ の部分集合 $\{m_j\}_{j \in \mathcal{J}}$ を表す。

2.2 ID ベース暗号

ID ベース暗号 Π は四つの PPT アルゴリズムの組 $(\mathsf{Setup}, \mathsf{KG}, \mathsf{Enc}, \mathsf{Dec})$ によって表される。以下、IBE の平文空間を $\mathcal{M}$、ID 空間を $\mathcal{ID}$ とする。セットアップアルゴリズム Setup は、セキュリティパラメータ 1^λ を入力として受け取り、公開パラメータ pp とマスター秘密鍵 msk を出力する。鍵生成アルゴリズム KG は、マスター秘密鍵 msk と ID $id \in \mathcal{ID}$ を入力として受け取り、ユーザ秘密鍵 sk_{id} を出力する。暗号化アルゴリズム Enc は、公開パラメータ pp、ID $id \in \mathcal{ID}$、平文 $m \in \mathcal{M}$ を入力として受け取り、暗号文 c を出力する。復号アルゴリズム Dec は、公開パラメータ pp、ユーザ秘密鍵 sk_{id}、暗号文 c を入力として受け取り、平文 $\widetilde{m} \in \{\bot\} \cup \mathcal{M}$ を出力する（$\bot$ は正しく復号できなかったことを表すシンボルである）。ID ベース暗号は正当性として、全ての $\lambda \in \mathbb{N}$、$m \in \mathcal{M}$、$id \in \mathcal{ID}$、$(pp, msk) \leftarrow \mathsf{Setup}(1^\lambda)$ に対して、$\mathsf{Dec}(pp, \mathsf{KG}(msk, id), \mathsf{Enc}(pp, id, m)) = m$ が成り立つことを要求する。

次に、ID ベース暗号に対する IND-ID-CPA 安全性を定義する。[*1]

定義 1 (IND-ID-CPA 安全性) $n := n(\lambda)$ を λ に関する多項式とする。ID ベース暗号方式 $\Pi = (\mathsf{Setup}, \mathsf{KG}, \mathsf{Enc}, \mathsf{Dec})$ に対して、$\mathcal{ID}$ と $\mathcal{M}$ をそれぞれ Π の ID 空間と平文空間とする。挑戦者と攻撃者 $\mathcal{A}$ に対して、IND-ID-CPA ゲームを以下の様に定義する。

(1) 挑戦者はチャレンジビット $b \leftarrow \{0,1\}$ を選択し、$(pp, msk) \leftarrow \mathsf{Setup}(1^\lambda)$ を生成する。次に、挑戦者は pp を $\mathcal{A}$ に送り、リスト $\mathsf{L}_{\mathsf{sk}} := \emptyset$ を用意する。
ゲーム中、$\mathcal{A}$ は次の鍵生成クエリを行うことができる。
鍵生成クエリ $\mathcal{A}$ は $id \in \mathcal{ID}$ を挑戦者に送る。挑戦者はまず、$(id, sk_{id}) \in \mathsf{L}_{\mathsf{sk}}$ かどうかをチェックする。$(id, sk_{id}) \in \mathsf{L}_{\mathsf{sk}}$ である場合、すでに生成された sk_{id} を $\mathcal{A}$ に返す。そうでないならば、$sk_{id} \leftarrow \mathsf{KG}(msk, id)$ を $\mathcal{A}$ に返し、(id, sk_{id}) を L_{sk} に追加する。

(2) $\mathcal{A}$ は $(id_i, m^*_{i,0}, m^*_{i,1})_{i \in [n]}$ を挑戦者に送る。（ただし、全ての $i \in [n]$ について $|m^*_{i,0}| = |m^*_{i,1}|$ かつ $(id_i, sk_{id_i}) \notin \mathsf{L}_{\mathsf{sk}}$ であるとする。）挑戦者は全ての $i \in [n]$ について $c^*_i \leftarrow \mathsf{Enc}(pp, id_i, m^*_{i,b})$ を計算し、$\mathbf{c}^* := (c^*_i)_{i \in [n]}$ を $\mathcal{A}$ に返す。以下、$\mathcal{A}$ は鍵生成クエリを行うことができるが、全ての $i \in [n]$ について id_i をクエリすることは禁止する。

(3) $\mathcal{A}$ は $b' \in \{0,1\}$ を出力する。

全ての PPT 攻撃者 $\mathcal{A}$ に対して、$\mathsf{Adv}^{\mathsf{ind\text{-}id\text{-}cpa}}_{\Pi,\mathcal{A}}(\lambda) := 2 \cdot |\Pr[b = b'] - \frac{1}{2}| = \mathsf{negl}(\lambda)$ が成り立つとき、Π は IND-ID-CPA 安全であるという。

2.3 非対話型ゼロ知識証明システム

$\mathcal{R}$ を効率的に計算可能な二値関係とし、組 $(x, w) \in \mathcal{R}$ に対して、x を命題、w を証拠と呼ぶ。また、NP 言語 $\mathcal{L}$ を $\mathcal{L} := \{x | \exists w : (x, w) \in \mathcal{R}\}$ で定義する。言語 $\mathcal{L}$ に対する非対話型ゼロ知識証明システムは、以下の五つの PPT アルゴリズムの組 $(\mathsf{CRSGen}, \mathsf{Prove}, \mathsf{Verify}, \mathsf{SimCRS}, \mathsf{SimPrv})$ で表される。共通参照情報 (CRS) 生成アルゴリズム CRSGen は、セキュリティパラメータ 1^λ を入力として受け取り、共通参照情報 crs を出力する。証明アルゴリズム Prove は、共通参照情報 crs、命題 $x \in \mathcal{L}$、及び、（$x \in \mathcal{L}$ に対する）証拠 w を入力として受け取り、証明 π を出力する。検証アルゴリズム Verify は、共通参照情報 crs、命題 x、及び、証明 π を入力として受け取り、1 または 0 を出力する。シミュレーション CRS 生成アルゴリズム SimCRS は、セキュリティパラメータ 1^λ を入力として受け取り、共通参照情報 crs と落とし戸情報 td を出力する。シミュレーション証明アルゴリズム SimPrv は、落とし戸情報 td と命題 x を入力として受け取り、証明 π を出力する。非対話型ゼロ知識証明システムは正当性として、全ての $\lambda \in \mathbb{N}$、共通参照情報 $crs \leftarrow \mathsf{CRSGen}(1^\lambda)$、命題 $x \in \mathcal{L}$、及び、（$x \in \mathcal{L}$ に対する）証拠 w に対して、$\mathsf{Verify}(crs, x, \mathsf{Prove}(crs, x, w)) = 1$ が成り立つことを要求する。

次に、非対話型ゼロ知識証明システムに対するシミュレーション健全性とゼロ知識性を定義する。[*2]

定義 2 (シミュレーション健全性) 非対話型ゼロ知識証明システム $\mathsf{PS} = (\mathsf{CRSGen}, \mathsf{Prove}, \mathsf{Verify}, \mathsf{SimCRS}, \mathsf{SimPrv})$ に対して、挑戦者と攻撃者 $\mathcal{A}$ の間の以下のゲームを考える。

(1) 挑戦者は、$(crs, td) \leftarrow \mathsf{SimCRS}(1^\lambda)$ を生成し、crs を $\mathcal{A}$ に送り、リスト $\mathsf{L} := \emptyset$ を用意する。ゲーム中、$\mathcal{A}$ は次のシミュレーションクエリを行うことができる。
シミュレーションクエリ $\mathcal{A}$ は命題 x を挑戦者に送る。挑戦者は、$\pi \leftarrow \mathsf{SimPrv}(td, x)$ を $\mathcal{A}$ に返し、(x, π) を L に追加する。

(2) $\mathcal{A}$ は命題と証明のペア (x^*, π^*) を出力する。

全ての PPT 攻撃者 $\mathcal{A}$ に対して、$\mathsf{Adv}^{\mathsf{s\text{-}sound}}_{\mathsf{PS},\mathcal{A}}(\lambda) := \Pr[x^* \notin \mathcal{L} \wedge \mathsf{Verify}(crs, x^*, \pi^*) = 1 \wedge (x^*, \pi^*) \notin \mathsf{L}] = \mathsf{negl}(\lambda)$ が成り立つとき、PS はシミュレーション健全性を満たすという。

定義 3 (ゼロ知識性) 非対話型ゼロ知識証明システム $\mathsf{PS} = (\mathsf{CRSGen}, \mathsf{Prove}, \mathsf{Verify}, \mathsf{SimCRS}, \mathsf{SimPrv})$ に対して、挑戦者と攻撃者 $\mathcal{A}$ の間の以下のゲームを考える。

(1) 挑戦者はまず、チャレンジビット $b \leftarrow \{0,1\}$ を選択する。$b = 0$ のとき、$crs \leftarrow \mathsf{CRSGen}(1^\lambda)$ を計算し、そうでないならば、$(crs, td) \leftarrow \mathsf{SimCRS}(1^\lambda)$ を計算する。挑戦者は、crs を $\mathcal{A}$ に送る。ゲーム中、$\mathcal{A}$ は

[*1] 本論文での証明に有用なため、IND-ID-CPA 安全性に対して、チャレンジを一回しか考えない通常の定義ではなく、複数回のチャレンジを考えた定義を導入する。本論文の定義と [6] での定義は、（アドバンテージの多項式倍の差を除いて）等価である。

[*2] 本論文では、非対話型ゼロ知識証明システムに対して、通常の健全性とゼロ知識性よりも強いシミュレーション健全性 “unbounded simulation soundness” とゼロ知識性 “multi-theorem zero-knowledge” を要求する。これらの要求を満たす非対話型ゼロ知識証明システムは、（通常の適応的安全な）非対話型ゼロ知識証明システムと一方向性関数から構成可能であることが知られている。 [18]

次の証明生成クエリを行うことができる。

証明生成クエリ $\mathcal{A}$ は 命題と証拠のペア (x, w) を挑戦者に送る。$b = 0$ のとき、挑戦者は $\pi \leftarrow \mathsf{Prove}(crs, x, w)$ を計算し、そうでないならば、$\pi \leftarrow \mathsf{SimPrv}(td, x)$ を計算する。そして、挑戦者は証明 π を $\mathcal{A}$ に返す。

(2) $\mathcal{A}$ は $b' \in \{0,1\}$ を出力する。

全ての PPT 攻撃者 $\mathcal{A}$ に対して、$\mathsf{Adv}^{\mathsf{zk}}_{\mathsf{PS},\mathcal{A}}(\lambda) := 2 \cdot |\Pr[b = b'] - \frac{1}{2}| = \mathsf{negl}(\lambda)$ が成り立つとき、PS はゼロ知識性を満たすという。

3. RSO-CCA 安全性を満たす ID ベース暗号

本章ではまず、ID ベース暗号に対する RSO-CCA 安全性を新たに導入する（3.1 節）。次に、IND-ID-CPA 安全性を満たす ID ベース暗号方式とシミュレーション健全性とゼロ知識性を満たす非対話型ゼロ知識証明システムを用いて、RSO-CCA 安全性を満たす ID ベース暗号方式を構成可能であることを示す（3.2 節）。

3.1 ID ベース暗号に対する RSO-CCA 安全性

本節では、ID ベース暗号に対する RSO-CCA 安全性を新たに導入する。

定義 4 (RSO-CCA 安全性) ID ベース暗号方式 $\Pi = (\mathsf{Setup}, \mathsf{KG}, \mathsf{Enc}, \mathsf{Dec})$ の ID 空間と平文空間をそれぞれ $\mathcal{ID}$ と $\mathcal{M}$ とする。$\mathcal{S}$ を PPT シミュレータとする。以下の挑戦者と攻撃者 $\mathcal{A}$ の間で行われるゲーム **RealGame**、及び、挑戦者とシミュレータ $\mathcal{S}$ の間で行われるゲーム **IdealGame** を考える。

RealGame.

(1) 挑戦者は、公開パラメータとマスター秘密鍵 $(pp, msk) \leftarrow \mathsf{Setup}(1^\lambda)$ を計算し、pp を $\mathcal{A}$ に送る。次に、挑戦者は鍵生成リスト $\mathsf{L} := \emptyset$ と鍵抽出リスト $\mathsf{L}_{\mathsf{ext}} := \emptyset$ を用意する。[*3] ゲーム中、$\mathcal{A}$ は以下の鍵生成クエリと復号クエリを行うことができる。

鍵生成クエリ $\mathcal{A}$ は $id \in \mathcal{ID}$ を挑戦者に送る。挑戦者はまず、$(id, sk_{id}) \in \mathsf{L}$ かどうかをチェックする。$(id, sk_{id}) \in \mathsf{L}$ である場合、すでに生成された sk_{id} を $\mathcal{A}$ に返し、(id, sk_{id}) を $\mathsf{L}_{\mathsf{ext}}$ に追加する。そうでないならば、$sk_{id} \leftarrow \mathsf{KG}(msk, id)$ を $\mathcal{A}$ に返し、(id, sk_{id}) を L と $\mathsf{L}_{\mathsf{ext}}$ に追加する。

復号クエリ $\mathcal{A}$ は (id, c) を挑戦者に送る。挑戦者はまず、$(id, sk_{id}) \in \mathsf{L}$ かどうかをチェックする。$(id, sk_{id}) \notin \mathsf{L}$ の場合のみ、挑戦者は id に対するユーザ秘密鍵 $sk_{id} \leftarrow \mathsf{KG}(msk, id)$ を生成し、(id, sk_{id}) を L に追加する。その後、挑戦者は $m \leftarrow \mathsf{Dec}(pp, sk_{id}, c)$ を計算して、m を $\mathcal{A}$ に返す。

(2) $\mathcal{A}$ は、n 個のチャレンジ ID $(id_i)_{i\in[n]}$ と $\mathcal{M}^n$ 上の平文分布 Dist を挑戦者に返す。(ただし、n は λ に関するあらかじめ制限されていない多項式であるとし、鍵生成クエリを行なった id をチャレンジ ID に含めることは禁止する。)
挑戦者は、チャレンジ平文 $(m^*_i)_{i\in[n]} \leftarrow \mathsf{Dist}$ をサンプルし、全ての $i \in [n]$ についてチャレンジ暗号文 $c^*_i \leftarrow \mathsf{Enc}(pp, id_i, m^*_i)$ を計算し、$\mathcal{A}$ に $\mathbf{c}^* := (c^*_i)_{i\in[n]}$ を返す。
このとき、挑戦者は全てのチャレンジ ID $(id_i)_{i\in[n]}$ に対して $(id_i, sk_{id_i}) \in \mathsf{L}$ かどうかをチェックする。$(id_i, sk_{id_i}) \notin \mathsf{L}$ の場合のみ、挑戦者は $sk_{id_i} \leftarrow \mathsf{KG}(msk, id_i)$ を計算して、L に (id_i, sk_{id_i}) を追加する。以下、$\mathcal{A}$ は鍵生成クエリと復号クエリを行うことはできるが、全ての $i \in [n]$ に対して、鍵生成クエリ id_i と復号クエリ (id_i, c^*_i) を行うことは禁止する。

(3) $\mathcal{A}$ は挑戦者に $\mathcal{J} \subset [n]$ を返す。挑戦者は、L 内の sk_{id_j} を用いて、$\mathbf{sk}^*_{\mathcal{J}} := (sk_{id_j})_{j\in\mathcal{J}}$ とセットして、$(\mathbf{m}^*_{\mathcal{J}}, \mathbf{sk}^*_{\mathcal{J}})$ を $\mathcal{A}$ に送る。以下、$\mathcal{A}$ は鍵生成クエリと復号クエリを行うことはできるが、上記での禁止制限に加えて、全ての $j \in \mathcal{J}$ に対して復号クエリ $(id_j, *)$ を行うことは禁止する。

(4) $\mathcal{A}$ は、out を挑戦者に返す。

(5) 挑戦者は、$\mathsf{out}_{\mathsf{real}} := ((id_i)_{i\in[n]}, (m^*_i)_{i\in[n]}, \mathsf{Dist}, \mathcal{J}, \mathsf{out})$ を出力する。

IdealGame.

(1) 挑戦者は、1^λ を $\mathcal{S}$ に送る。

(2) $\mathcal{S}$ は、n 個のチャレンジ ID $(id_i)_{i\in[n]}$ と $\mathcal{M}^n$ 上の平文分布 Dist を挑戦者に返す。(ただし、n は λ に関するあらかじめ制限されていない多項式であるとする。) 挑戦者は、チャレンジ平文 $(m^*_i)_{i\in[n]} \leftarrow \mathsf{Dist}$ をサンプルする。

(3) $\mathcal{S}$ は、$\mathcal{J} \subset [n]$ を挑戦者に送る。挑戦者は、$(m^*_j)_{j\in\mathcal{J}}$ を $\mathcal{S}$ に返す。

(4) $\mathcal{S}$ は out を挑戦者に返す。

(5) 挑戦者は、$\mathsf{out}_{\mathsf{ideal}} := ((id_i)_{i\in[n]}, (m^*_i)_{i\in[n]}, \mathsf{Dist}, \mathcal{J}, \mathsf{out})$ を出力する。

全ての PPT 攻撃者 $\mathcal{A}$ に対して、ある PPT シミュレータ $\mathcal{S}$ が存在して、全ての PPT 識別者 $\mathcal{D}$ に対して、$\mathsf{Adv}^{\mathsf{rso\text{-}cca}}_{\Pi,\mathcal{A},\mathcal{S},\mathcal{D}}(\lambda) := |\Pr[\mathcal{D}(\mathsf{out}_{\mathsf{real}}) = 1] - \Pr[\mathcal{D}(\mathsf{out}_{\mathsf{ideal}}) = 1]| = \mathsf{negl}(\lambda)$ が成り立つとき、Π は RSO-CCA 安全であるという。

注意 1 簡単のため、上記の試行では攻撃者による非適応的な秘密鍵のコラプトのみを考える。すなわち、インデックス集合 $\mathcal{J}$ をまとめて一度だけ出力するように定義する。しかしながら、我々の提案方式は攻撃者による適応的な秘密鍵のコラプトを考慮したとしても、その安全性を保証できる。

3.2 RSO-CCA 安全な ID ベース暗号の構成方法

本節では、IND-ID-CPA 安全な ID ベース暗号方式とシミュレーション健全性とゼロ知識性を満たす非対話型ゼロ知識証明システムを用いて、RSO-CCA 安全な ID ベース暗号方式を構成可能であることを示す。$\Pi = (\mathsf{Setup}, \mathsf{KG}, \mathsf{Enc}, \mathsf{Dec})$ を平文空間と ID 空間がそれぞれ $\{0,1\}$ と $\mathcal{ID}\times\{0,1\}$ であるような ID ベース暗号方式であるとし、Enc の乱数空間を $\mathcal{R}_\Pi$ とする。また、$\mathsf{PS} = (\mathsf{CRSGen}, \mathsf{Prove}, \mathsf{Verify}, \mathsf{SimCRS}, \mathsf{SimPrv})$ を NP 言語

$$\mathcal{L}_{eq} := \{(pp, id, c_0, c_1) | \exists (m, r_0, r_1)\ s.t.$$
$$c_0 = \mathsf{Enc}(pp, (id, 0), m; r_0) \wedge c_1 = \mathsf{Enc}(pp, (id, 1), m; r_1)\}$$

に対する非対話型ゼロ知識証明システムとする。このとき、平文空間と ID 空間がそれぞれ $\{0,1\}$ と $\mathcal{ID}$ であるような ID ベース暗号方式 $\Pi' = (\mathsf{Setup}', \mathsf{KG}', \mathsf{Enc}', \mathsf{Dec}')$ を以下のように構成する。

$\mathsf{Setup}'(1^\lambda)$: $(pp, msk) \leftarrow \mathsf{Setup}(1^\lambda)$, $crs \leftarrow \mathsf{CRSGen}(1^\lambda)$ を計算する。$PP := (pp, crs)$, $MSK := msk$ とセットして、(PP, MSK) を出力する。

$\mathsf{KG}'(MSK, id)$: α を $\{0,1\}$ から一様ランダムに選び、$sk_{id,\alpha} \leftarrow \mathsf{KG}(msk, (id, \alpha))$ を計算する。$SK_{id} := (\alpha, sk_{id,\alpha}, id)$ とセットして、SK_{id} を出力する。

$\mathsf{Enc}'(PP, id, m)$: r_0 と r_1 を $\mathcal{R}_\Pi$ から一様ランダムにサンプルし、$c_0 \leftarrow \mathsf{Enc}(pp, (id, 0), m; r_0)$,

[*3] 鍵生成リスト L は、ゲーム中に生成された全てのユーザ秘密鍵を保存しておくためのリストである。一方で、鍵抽出リスト $\mathsf{L}_{\mathsf{ext}}$ は、鍵生成クエリで攻撃者に渡されたユーザ秘密鍵のみを保存しておくためのリストである。よって、常に $\mathsf{L}_{\mathsf{ext}} \subset \mathsf{L}$ が成り立つ。

$c_1 \leftarrow \mathsf{Enc}(pp,(id,1),m;r_1)$, $\pi \leftarrow \mathsf{Prove}(crs,(pp,id,c_0,c_1),(m,r_0,r_1))$ を計算する。$c:=(c_0,c_1,\pi)$ とセットし、c を出力する。

$\mathsf{Dec}'(PP,SK_{id},c)$: $\mathsf{Verify}(crs,(pp,id,c_0,c_1),\pi)=1$ が成り立つならば、$m \leftarrow \mathsf{Dec}(pp,sk_{id,\alpha},c_\alpha)$ を計算して、m を出力する。そうでなければ、$\perp$ を出力する。

このとき、以下の定理が成り立つ。

定理 1 Π が IND-ID-CPA 安全な ID ベース暗号方式であり、かつ、PS がシミュレーション健全性とゼロ知識性を満たす非対話型ゼロ知識証明システムであるならば、ID ベース暗号方式 Π' は RSO-CCA 安全性を満たす。

定理 1 の証明

$\mathcal{A}$ を Π' の RSO-CCA 安全性に対する任意の攻撃者であるとする。このとき、$\mathcal{A}$ に対するシミュレータ $\mathcal{S}$ を次のように構成する。

(1) $\mathcal{S}$ は 1^λ を受け取り、$(pp,msk) \leftarrow \mathsf{Setup}(1^\lambda)$ と $(crs,td) \leftarrow \mathsf{SimCRS}(1^\lambda)$ を計算し、自身の鍵生成リスト $\mathsf{L}^{\mathcal{S}}:=\emptyset$ と鍵抽出リスト $\mathsf{L}^{\mathcal{S}}_{\mathsf{ext}}:=\emptyset$ を用意する。そして、$\mathcal{S}$ は $PP:=(pp,crs)$ とセットして $\mathcal{A}$ に PP を送る。

鍵生成クエリ $\mathcal{A}$ が id を送ってきたら、$\mathcal{S}$ はまず $(id,SK'_{id}=(\perp,sk_{id,0},sk_{id,1},id)) \in \mathsf{L}^{\mathcal{S}} \backslash \mathsf{L}^{\mathcal{S}}_{\mathsf{ext}}$ かどうかをチェックする。$(id,SK'_{id}) \in \mathsf{L}^{\mathcal{S}} \backslash \mathsf{L}^{\mathcal{S}}_{\mathsf{ext}}$ の場合のみ、$\mathcal{S}$ は α を $\{0,1\}$ から一様ランダムにサンプルし、$SK'_{id}:=(\alpha,sk_{id,0},sk_{id,1},id)$ と更新する。
その後、$\mathcal{S}$ は $(id,SK'_{id}) \in \mathsf{L}^{\mathcal{S}}$ かどうかをチェックする。$(id,SK'_{id}) \in \mathsf{L}^{\mathcal{S}}$ である場合、$\mathcal{S}$ は既に生成された $SK'_{id}=(\alpha,sk_{id,0},sk_{id,1},id)$ を用いて、$SK_{id}:=(\alpha,sk_{id,\alpha},id)$ を $\mathcal{A}$ に返し、$\mathsf{L}^{\mathcal{S}}_{\mathsf{ext}}$ に (id,SK'_{id}) を追加する。そうでないならば、$\mathcal{S}$ は α を $\{0,1\}$ から一様ランダムにサンプルし、$sk_{id,0} \leftarrow \mathsf{KG}(msk,(id,0))$ と $sk_{id,1} \leftarrow \mathsf{KG}(msk,(id,1))$ を計算する。その後、$\mathcal{S}$ は $SK_{id}:=(\alpha,sk_{id,\alpha},id)$ を $\mathcal{A}$ に返し、$SK'_{id}:=(\alpha,sk_{id,0},sk_{id,1},id)$ とセットして、$\mathsf{L}^{\mathcal{S}}$ と $\mathsf{L}^{\mathcal{S}}_{\mathsf{ext}}$ に (id,SK'_{id}) を追加する。
復号クエリ $\mathcal{A}$ が $(id,c=(c_0,c_1,\pi))$ を送ってきたら、$\mathcal{S}$ はまず $\mathsf{Verify}(crs,(pp,id,c_0,c_1),\pi)$ を計算する。もし $\mathsf{Verify}(crs,(pp,id,c_0,c_1),\pi)=0$ であるならば、$\mathcal{S}$ は $\perp$ を $\mathcal{A}$ に返す。そうでないならば、$\mathcal{S}$ は次に $(id,SK'_{id}) \in \mathsf{L}^{\mathcal{S}}$ かどうかをチェックする。$(id,SK'_{id}) \notin \mathsf{L}^{\mathcal{S}}$ の場合のみ、$\mathcal{S}$ は $sk_{id,0} \leftarrow \mathsf{KG}(msk,(id,0))$ と $sk_{id,1} \leftarrow \mathsf{KG}(msk,(id,1))$ を計算する。そして、$\mathcal{S}$ は $SK'_{id}:=(\perp,sk_{id,0},sk_{id,1},id)$ とセットして、(id,SK'_{id}) を $\mathsf{L}^{\mathcal{S}}$ に追加する。その後、$\mathcal{S}$ は $m \leftarrow \mathsf{Dec}(pp,sk_{id,0},c_0)$ を計算し、m を $\mathcal{A}$ に返す。

(2) $\mathcal{A}$ が n 個のチャレンジ ID $(id_i)_{i\in[n]}$ と平文分布 Dist を送ってきたら、$\mathcal{S}$ はこれらを挑戦者に送る。そして、$\mathcal{S}$ はまず全ての $i \in [n]$ に対して、$(id_i,SK'_{id_i}=(\perp,sk_{id_i,0},sk_{id_i,1},id_i)) \in \mathsf{L}^{\mathcal{S}} \backslash \mathsf{L}^{\mathcal{S}}_{\mathsf{ext}}$ かどうかをチェックする。$(id_i,SK'_{id_i}) \in \mathsf{L}^{\mathcal{S}} \backslash \mathsf{L}^{\mathcal{S}}_{\mathsf{ext}}$ の場合のみ、$\mathcal{S}$ は α_i を $\{0,1\}$ から一様ランダムにサンプルし、$SK'_{id_i}:=(\alpha_i,sk_{id_i,0},sk_{id_i,1},id_i)$ と更新する。その後、$\mathcal{S}$ は全ての $i \in [n]$ に対して、$(id_i,SK'_{id_i}) \in \mathsf{L}^{\mathcal{S}}$ かどうかをチェックする。$(id_i,SK'_{id_i}) \notin \mathsf{L}^{\mathcal{S}}$ の場合のみ、$\mathcal{S}$ は α_i を $\{0,1\}$ から一様ランダムにサンプルし、$sk_{id_i,0} \leftarrow \mathsf{KG}(msk,(id_i,0))$ と $sk_{id_i,1} \leftarrow \mathsf{KG}(msk,(id_i,1))$ を計算する。そして、$\mathcal{S}$ は $SK'_{id_i}:=(\alpha_i,sk_{id_i,0},sk_{id_i,1},id_i)$ とセットして、(id_i,SK'_{id_i}) を $\mathsf{L}^{\mathcal{S}}$ に追加する。その後、$\mathcal{S}$ は全ての $i \in [n]$ に対して、SK'_{id_i} 内の $\alpha_i \in \{0,1\}$ に基づいて、$c^*_{i,\alpha_i} \leftarrow \mathsf{Enc}(pp,(id_i,\alpha_i),0)$ と $c^*_{i,1\oplus\alpha_i} \leftarrow \mathsf{Enc}(pp,(id_i,1\oplus\alpha_i),1)$ を計算する。そして、$\mathcal{S}$ は全ての $i \in [n]$ に対して、$x^*_i:=(pp,id_i,c^*_{i,0},c^*_{i,1})$ とセットして $\pi^*_i \leftarrow \mathsf{SimPrv}(td,x^*_i)$ を計算して、$c^*_i:=(c^*_{i,0},c^*_{i,1},\pi^*_i)$ とセットする。

最後に、$\mathcal{S}$ は $\mathbf{c}^*:=(c^*_i)_{i\in[n]}$ を $\mathcal{A}$ に返す。$\mathcal{S}$ は $\mathcal{A}$ からの鍵生成クエリと復号クエリに対して、(1) のときと同様に答える。

(3) $\mathcal{A}$ が $\mathcal{J} \subset [n]$ を送ってきたら、$\mathcal{S}$ は $\mathcal{J}$ を挑戦者に送る。$\mathcal{S}$ は $\mathbf{m}^*_{\mathcal{J}}$ を受け取ったら、全ての $j \in \mathcal{J}$ に対して $\mathsf{L}^{\mathcal{S}}$ 内の SK_{id_j} を用いて、$\widetilde{SK}_{id_j}:=(\alpha_j \oplus m^*_j, sk_{id_j,\alpha_j\oplus m^*_j}, id_j)$ とセットする。そして、$\mathcal{S}$ は $\mathbf{SK}^*_{\mathcal{J}}:=(\widetilde{SK}_{id_j})_{j\in\mathcal{J}}$ とセットして、$\mathcal{A}$ に $(\mathbf{m}^*_{\mathcal{J}},\mathbf{SK}^*_{\mathcal{J}})$ を返す。$\mathcal{S}$ は $\mathcal{A}$ からの鍵生成クエリと復号クエリに対して、(1) のときと同様に答える。

(4) $\mathcal{A}$ が out を出力したら、$\mathcal{S}$ は out を挑戦者に返す。

任意の攻撃者 $\mathcal{A}$ と上記のシミュレータ $\mathcal{S}$ に対して、$\mathcal{D}$ を任意の PPT 識別者とする。以下の証明中では、**RealGame** 中で挑戦者が出力した $\mathsf{out}_{\mathsf{real}}$ を受け取った識別者 $\mathcal{D}$ が 1 を出力する確率を p_{real} とする。また、**IdealGame** 中で挑戦者が出力した $\mathsf{out}_{\mathsf{ideal}}$ を受け取った識別者 $\mathcal{D}$ が 1 を出力する確率を p_{ideal} とする。ここで以下のゲーム列 $\{\mathbf{Game}_i\}_{i=0}^{5}$ を導入する。

$\mathbf{Game}_0$: **RealGame** と同じゲームである。

$\mathbf{Game}_1$: PS の共通参照情報 crs を SimCRS によって生成されるものに変更する。さらに、全てのチャレンジ ID $(id_i)_{i\in[n]}$ について、チャレンジ暗号文 c^*_i を計算する際に、PS の証明 π^*_i を SimPrv によって生成されるものに変更する。上記の変更を除いて、$\mathbf{Game}_1$ は $\mathbf{Game}_0$ と同じゲームである。

$\mathbf{Game}_2$: 全てのチャレンジ ID $(id_i)_{i\in[n]}$ について、チャレンジ暗号文 c^*_i を計算する際に、$c^*_{i,1\oplus\alpha_i} \leftarrow \mathsf{Enc}(pp,(id_i,1\oplus\alpha_i),m^*_i)$ を $c^*_{i,1\oplus\alpha_i} \leftarrow \mathsf{Enc}(pp,(id_i,1\oplus\alpha_i),1\oplus m^*)$ に変更する。上記の変更を除いて、$\mathbf{Game}_2$ は $\mathbf{Game}_1$ と同じゲームである。

$\mathbf{Game}_3$: 復号クエリ $(id,c=(c_0,c_1,\pi))$ に答える際に、$\mathsf{Verify}(crs,(pp,id,c_0,c_1),\pi)=1$ であるならば、$m \leftarrow \mathsf{Dec}(pp,sk_{id,\alpha},c_\alpha)$ の代わりに $m \leftarrow \mathsf{Dec}(pp,sk_{id,0},c_0)$ で答えるように変更する。

より詳細には、挑戦者は $\mathcal{A}$ からの復号クエリ (id,c) に対して、$\mathsf{Verify}(crs,(pp,id,c_0,c_1),\pi)=1$ であるときに以下の通りに対応するように変更する。まず挑戦者は、$(id,SK_{id}) \in \mathsf{L}$ であるかどうかをチェックする。$(id,SK_{id}) \notin \mathsf{L}$ の場合のみ、挑戦者は、($\alpha \leftarrow \{0,1\}$ は行わずに) $sk_{id,0} \leftarrow \mathsf{KG}(msk,(id,0))$ と $sk_{id,1} \leftarrow \mathsf{KG}(msk,(id,1))$ を計算し、$SK_{id}:=(\perp,sk_{id,0},sk_{id,1},id)$ とセットして、(id,SK_{id}) を L に追加する。この後に挑戦者は、$m \leftarrow \mathsf{Dec}(pp,sk_{id,0},c_0)$ を計算し、m を $\mathcal{A}$ に返す。

この変更により、復号クエリの応答時に、鍵生成クエリが行われていない id に対する SK_{id} に対しては、α を選ばないように変更された。そこでこのゲームより、チャレンジ ID id_i に関する復号クエリ (id_i,c) が行われていた場合、id_i に対するチャレンジ暗号文 c^*_i を計算する際に、α_i だけを生成するように変更する。

上記の変更を除いて、$\mathbf{Game}_3$ は $\mathbf{Game}_2$ と同じゲームである。

$\mathbf{Game}_4$: 全てのチャレンジ ID $(id_i)_{i\in[n]}$ について、α_i の代わりに $\alpha_i \oplus m_i^*$ を用いるように変更する。すなわち、チャレンジ暗号文 c_i^* を計算する際に、$c_{i,0}^*$ と $c_{i,1}^*$ はそれぞれ、$c_{i,\alpha_i\oplus m_i^*}^* \leftarrow \mathsf{Enc}(pp,(id_i,\alpha_i\oplus m_i^*),m_i^*)$ と $c_{i,\alpha_i\oplus(1\oplus m_i^*)}^* \leftarrow \mathsf{Enc}(pp,(id_i,\alpha_i\oplus(1\oplus m_i^*)),1\oplus m_i^*)$ と計算するように変更する。また、$\mathcal{A}$ に与える id_i に対するユーザ秘密鍵は、$\widetilde{SK}_{id_i} := (\alpha_i\oplus m_i^*, sk_{id_i,\alpha_i\oplus m_i^*}, id_i)$ に変更する。上記の変更を除いて、$\mathbf{Game}_4$ は $\mathbf{Game}_3$ と同じゲームである。

$\mathbf{Game}_5$: **IdealGame** と同じゲームである。

各 $i\in[0,5]$ について、out_i' を $\mathbf{Game}_i$ で挑戦者が最終的に出力する値とし、$\mathbf{T}_i$ を $\mathcal{D}$ が挑戦者からの出力結果 out_i' を受け取って 1 を出力する事象とする。このとき、$p_{\mathsf{real}} = \Pr[\mathbf{T}_0]$ であり、かつ、$p_{\mathsf{ideal}} = \Pr[\mathbf{T}_5]$ であるため、$\mathsf{Adv}^{\mathsf{rso\text{-}cca}}_{\Pi,\mathcal{A},\mathcal{S},\mathcal{D}}(\lambda)$ を以下の様に見積もることができる。

$$\begin{aligned}\mathsf{Adv}^{\mathsf{rso\text{-}cca}}_{\Pi,\mathcal{A},\mathcal{S},\mathcal{D}}(\lambda) &= |\Pr[\mathbf{T}_0]-\Pr[\mathbf{T}_5]|\\ &\le \sum_{i=0}^{4}\left|\Pr[\mathbf{T}_i]-\Pr[\mathbf{T}_{i+1}]\right|\end{aligned}$$

以下では、$|\Pr[\mathbf{T}_0]-\Pr[\mathbf{T}_1]| = \mathsf{Adv}^{\mathsf{zk}}_{\mathsf{PS},\mathcal{B}_1}(\lambda)$ を満たすゼロ知識性に対する攻撃者 $\mathcal{B}_1$, $|\Pr[\mathbf{T}_1]-\Pr[\mathbf{T}_2]| = \mathsf{Adv}^{\mathsf{ind\text{-}id\text{-}cpa}}_{\Pi,\mathcal{B}_2}(\lambda)$ を満たす IND-ID-CPA 安全性に対する攻撃者 $\mathcal{B}_2$, $|\Pr[\mathbf{T}_2]-\Pr[\mathbf{T}_3]| \le \mathsf{Adv}^{\mathsf{s\text{-}sound}}_{\mathsf{PS},\mathcal{B}_3}(\lambda)$ を満たすシミュレーション健全性に対する攻撃者 $\mathcal{B}_3$ がそれぞれ存在することを補題として示す。また、$|\Pr[\mathbf{T}_3]-\Pr[\mathbf{T}_4]| = |\Pr[\mathbf{T}_4]-\Pr[\mathbf{T}_5]| = 0$ が成り立つことを補題として示す。

補題 1 $|\Pr[\mathbf{T}_0]-\Pr[\mathbf{T}_1]| = \mathsf{Adv}^{\mathsf{zk}}_{\mathsf{PS},\mathcal{B}_1}(\lambda)$ を満たす PPT 攻撃者 $\mathcal{B}_1$ が存在する。

補題 1 の証明

Π' に対する攻撃者 $\mathcal{A}$ を用いて、PS のゼロ知識性に対する攻撃者 $\mathcal{B}_1$ を以下のように構成する。

(1) $\mathcal{B}_1$ は挑戦者から crs を受け取り、$(pp,msk) \leftarrow \mathsf{Setup}(1^\lambda)$ を計算して、$PP := (pp, crs)$ とセットして、$\mathcal{A}$ に PP を送る。$\mathcal{B}_1$ は、自分の鍵生成リスト $\mathsf{L}^{\mathcal{B}_1} := \emptyset$ と鍵抽出リスト $\mathsf{L}^{\mathcal{B}_1}_{\mathsf{ext}} := \emptyset$ を用意する。

鍵生成クエリ $\mathcal{A}$ の鍵生成クエリ id に対して、$\mathcal{B}_1$ はまず $(id,SK_{id}) \in \mathsf{L}^{\mathcal{B}_1}$ かどうかをチェックする。$(id,SK_{id}) \in \mathsf{L}^{\mathcal{B}_1}$ である場合、$\mathcal{B}_1$ は既に生成された SK_{id} を $\mathcal{A}$ に返し、$\mathsf{L}^{\mathcal{B}_1}_{\mathsf{ext}}$ に (id,SK_{id}) を追加する。そうでないならば、$\mathcal{B}_1$ は $SK_{id} \leftarrow \mathsf{KG}'(msk,id)$ を $\mathcal{A}$ に返し、$\mathsf{L}^{\mathcal{B}_1}$ と $\mathsf{L}^{\mathcal{B}_1}_{\mathsf{ext}}$ に (id,SK_{id}) を追加する。

復号クエリ $\mathcal{A}$ の復号クエリ (id,c) に対して、$\mathcal{B}_1$ はまず $(id,SK_{id}) \in \mathsf{L}^{\mathcal{B}_1}$ かどうかをチェックする。$(id,SK_{id}) \notin \mathsf{L}^{\mathcal{B}_1}$ の場合のみ、$\mathcal{B}_1$ は id に対するユーザ秘密鍵 $SK_{id} \leftarrow \mathsf{KG}'(msk,id)$ を計算して、$\mathsf{L}^{\mathcal{B}_1}$ に (id,SK_{id}) を追加する。その後、$\mathcal{B}_1$ は $m \leftarrow \mathsf{Dec}'(PP,SK_{id},c)$ を計算して、m を $\mathcal{A}$ に返す。

(2) $\mathcal{B}_1$ は $\mathcal{A}$ から n 個のチャレンジ ID $(id_i)_{i\in[n]}$ と平文分布 Dist を受け取り、まずチャレンジ平文 $(m_i^*) \leftarrow \mathsf{Dist}$ をサンプルし、全ての $i\in[n]$ について $(r_{i,0}^*, r_{i,1}^*) \leftarrow \mathcal{R}_\Pi^2$ をサンプルし、$c_{i,0}^* \leftarrow \mathsf{Enc}(pp,(id_i,0),m_i^*;r_{i,0}^*)$, $c_{i,1}^* \leftarrow \mathsf{Enc}(pp,(id_i,1),m_i^*;r_{i,1}^*)$ を計算する。次に、$\mathcal{B}_1$ は全ての $i\in[n]$ について $x_i^* := (pp,id_i,c_{i,0}^*,c_{i,1}^*)$, $w_i^* := (m_i^*,r_{i,0}^*,r_{i,1}^*)$ とセットして、挑戦者に (x_i^*,w_i^*) をクエリする。そして、$\mathcal{B}_1$ は挑戦者から証明 π_i^* を受け取り、$c_i^* := (c_{i,0}^*,c_{i,1}^*,\pi_i^*)$ とセットして、$\mathcal{A}$ に $\mathbf{c}^* := (c_i^*)_{i\in[n]}$ を返す。

このとき、$\mathcal{B}_1$ は全てのチャレンジ ID $(id_i)_{i\in[n]}$ に対して $(id_i,SK_{id_i}) \in \mathsf{L}^{\mathcal{B}_1}$ かどうかをチェックする。$(id_i,SK_{id_i}) \notin \mathsf{L}^{\mathcal{B}_1}$ である場合、$\mathcal{B}_1$ は $SK_{id_i} \leftarrow \mathsf{KG}'(msk,id_i)$ を計算して、$\mathsf{L}^{\mathcal{B}_1}$ に (id_i,SK_{id_i}) を追加する。$\mathcal{B}_1$ は $\mathcal{A}$ からの鍵生成クエリと復号クエリに対して、(1) のときと同様に答える。

(3) $\mathcal{B}_1$ は、$\mathcal{A}$ からインデックス集合 $\mathcal{J}\subset[n]$ を受け取り、$\mathsf{L}^{\mathcal{B}_1}$ 内の $(SK_{id_j})_{j\in\mathcal{J}}$ を用いて $\mathbf{SK}^*_{\mathcal{J}} := (SK_{id_j})_{j\in\mathcal{J}}$ とセットして、$\mathbf{SK}^*_{\mathcal{J}}$ を $\mathcal{A}$ に返す。$\mathcal{B}_1$ は $\mathcal{A}$ からの鍵生成クエリと復号クエリに対して、(1) のときと同様に答える。

(4) $\mathcal{B}_1$ は、$\mathcal{A}$ から受け取った out を用いて $\mathsf{out}' := ((id_i)_{i\in[n]},(m_i^*)_{i\in[n]},\mathsf{Dist},\mathcal{J},\mathsf{out})$ とセットして、$b' \leftarrow \mathcal{D}(\mathsf{out}')$ を計算して $\beta' := b'$ として挑戦者に返す。

$\mathcal{B}_1$ と挑戦者の間で行われるゲームのチャレンジビットを β とする。$\beta=0$ のとき、$\mathcal{B}_1$ は挑戦者から本物の CRS crs と全ての $i\in[n]$ について本物の証明 $\pi_i^* \leftarrow \mathsf{Prove}(crs,x_i^*,w_i^*)$ を受け取る。よって、$\mathcal{B}_1$ は $\mathcal{A}$ に対して $\mathbf{Game}_0$ を完全にシミュレートしているため、$\mathcal{D}$ に入力される out' の分布は、$\mathbf{Game}_0$ での分布と完全に同一である。一方で、$\beta=1$ のとき、$\mathcal{B}_1$ は挑戦者からシミュレートされた CRS crs と全ての $i\in[n]$ についてシミュレートされた証明 $\pi_i^* \leftarrow \mathsf{SimPrv}(td,x_i^*)$ を受け取る。よって、$\mathcal{B}_1$ は $\mathcal{A}$ に対して $\mathbf{Game}_1$ を完全にシミュレートしているため、$\mathcal{D}$ に入力される out' の分布は、$\mathbf{Game}_1$ での分布と完全に同一である。したがって、$\mathsf{Adv}^{\mathsf{zk}}_{\mathsf{PS},\mathcal{B}_1}(\lambda) = 2\cdot|\Pr[\beta=\beta']-\frac{1}{2}| = |\Pr[\beta'=1|\beta=0]-\Pr[\beta'=1|\beta=1]| = |\Pr[\mathbf{T}_0]-\Pr[\mathbf{T}_1]|$ が成り立つ。 □ **(補題 1)**

補題 2 $|\Pr[\mathbf{T}_1]-\Pr[\mathbf{T}_2]| = \mathsf{Adv}^{\mathsf{ind\text{-}id\text{-}cpa}}_{\Pi,\mathcal{B}_2}(\lambda)$ を満たす PPT 攻撃者 $\mathcal{B}_2$ が存在する。

補題 2 の証明

Π' に対する攻撃者 $\mathcal{A}$ を用いて、Π の IND-ID-CPA 安全性に対する攻撃者 $\mathcal{B}_2$ を以下のように構成する。

(1) $\mathcal{B}_2$ は挑戦者から pp を受け取り、$(crs,td) \leftarrow \mathsf{SimCRS}(1^\lambda)$ を計算して、$PP := (pp,crs)$ とセットして、$\mathcal{A}$ に PP を送る。$\mathcal{B}_2$ は、自分の鍵生成リスト $\mathsf{L}^{\mathcal{B}_2} := \emptyset$ と鍵抽出リスト $\mathsf{L}^{\mathcal{B}_2}_{\mathsf{ext}} := \emptyset$ を用意する。

鍵生成クエリ $\mathcal{A}$ の鍵生成クエリ id に対して、$\mathcal{B}_2$ はまず $(id,SK_{id}) \in \mathsf{L}^{\mathcal{B}_2}$ かどうかをチェックする。$(id,SK_{id}) \in \mathsf{L}^{\mathcal{B}_2}$ である場合、既に生成された SK_{id} を $\mathcal{A}$ に返し、$\mathsf{L}^{\mathcal{B}_2}_{\mathsf{ext}}$ に (id,SK_{id}) を追加する。そうでない場合、α を $\{0,1\}$ から一様ランダムにサンプルし、挑戦者に鍵生成クエリ (id,α) を行う。$\mathcal{B}_2$ は挑戦者から $sk_{id,\alpha}$ を受け取り、$SK_{id} := (\alpha, sk_{id,\alpha}, id)$ を $\mathcal{A}$ に返し、$\mathsf{L}^{\mathcal{B}_2}$ と $\mathsf{L}^{\mathcal{B}_2}_{\mathsf{ext}}$ に (id,SK_{id}) を追加する。

復号クエリ $\mathcal{A}$ の復号クエリ (id,c) に対して、$\mathcal{B}_2$ はまず $c = (c_0,c_1,\pi)$ に分割し、$\mathsf{Verify}(crs,(pp,id,c_0,c_1),\pi) = 1$ が成り立つかどうかを検証する。成り立たないならば、$\mathcal{B}_2$ は、$\perp$ を $\mathcal{A}$ に返す。成り立つならば、$\mathcal{B}_2$ は次に $(id,SK_{id}) \in \mathsf{L}^{\mathcal{B}_2}$ かどうかをチェックする。$(id,SK_{id}) \notin \mathsf{L}^{\mathcal{B}_2}$ の場合のみ、$\mathcal{B}_2$ は α を $\{0,1\}$ から一様ランダムにサンプルし、挑戦者に鍵生成クエリ (id,α) を行う。$\mathcal{B}_2$ は挑戦者から $sk_{id,\alpha}$ を受け取り、$SK_{id} := (\alpha, sk_{id,\alpha}, id)$ とセットして、(id,SK_{id}) を $\mathsf{L}^{\mathcal{B}_2}$ に追加する。その後、$\mathcal{B}_2$ は SK_{id} 内の $sk_{id,\alpha}$ を用いて、$m \leftarrow \mathsf{Dec}(pp,sk_{id,\alpha},c)$ を $\mathcal{A}$ に返す。

(2) $\mathcal{B}_2$ は $\mathcal{A}$ から n 個のチャレンジ ID $(id_i)_{i\in[n]}$ と平文分布

Dist を受け取り、まずチャレンジ平文 $(m_i^*)_{i\in[n]} \leftarrow \mathsf{Dist}$ をサンプルして、全てのチャレンジ ID $(id_i)_{i\in[n]}$ に対して $(id_i, SK_{id_i}) \in \mathsf{L}^{\mathcal{B}_2}$ かどうかをチェックする。$(id_i, SK_{id_i}) \notin \mathsf{L}^{\mathcal{B}_2}$ の場合のみ、$\mathcal{B}_2$ は α_i を $\{0,1\}$ から一様ランダムにサンプルし、挑戦者に鍵生成クエリ (id_i, α_i) を行う。$\mathcal{B}_2$ は挑戦者から sk_{id_i,α_i} を受け取る。その後、$\mathcal{B}_2$ は $SK_{id_i} := (\alpha_i, sk_{id_i,\alpha_i}, id_i)$ とセットして、(id_i, SK_{id_i}) を $\mathsf{L}^{\mathcal{B}_2}$ に追加する。

その後、$\mathcal{B}_2$ はチャレンジ $((id_i, 1\oplus\alpha_i), m_i^*, 1\oplus m_i^*)_{i\in[n]}$ を挑戦者に送り、$(c^*_{i,1\oplus\alpha_i})_{i\in[n]}$ を受け取る。そして、$\mathcal{B}_2$ は全ての $i \in [n]$ に対して、$c^*_{i,\alpha_i} \leftarrow \mathsf{Enc}(pp, (id_i, \alpha_i), m_i^*)$ を計算して、$x_i^* := (pp, id_i, c^*_{i,0}, c^*_{i,1})$ とセットして、$\pi_i^* \leftarrow \mathsf{SimPrv}(td, x_i^*)$ を計算し、$c_i^* := (c^*_{i,0}, c^*_{i,1}, \pi_i^*)$ とセットする。最後に $\mathcal{B}_2$ は $\mathcal{A}$ に $\mathbf{c}^* := (c_i^*)_{i\in[n]}$ を返す。$\mathcal{B}_2$ は $\mathcal{A}$ からの鍵生成クエリと復号クエリに対して、(1) のときと同様に答える。

(3) $\mathcal{B}_2$ は、$\mathcal{A}$ からインデックス集合 $\mathcal{J} \subset [n]$ を受け取り、$\mathsf{L}^{\mathcal{B}_2}$ 内の $(SK_{id_j})_{j\in\mathcal{J}}$ を用いて $\mathbf{SK}^*_{\mathcal{J}} := (SK_{id_j})_{j\in\mathcal{J}}$ とセットして、$\mathbf{SK}^*_{\mathcal{J}}$ を $\mathcal{A}$ に返す。$\mathcal{B}_2$ は $\mathcal{A}$ からの鍵生成クエリと復号クエリに対して、(1) のときと同様に答える。

(4) $\mathcal{B}_2$ は、$\mathcal{A}$ から受け取った out を用いて、$\mathsf{out}' := ((id_i)_{i\in[n]}, (m_i^*)_{i\in[n]}, \mathsf{Dist}, \mathcal{J}, \mathsf{out})$ とセットして、$b' \leftarrow \mathcal{D}(\mathsf{out}')$ を計算して $\beta' := b'$ として挑戦者に返す。

以上が $\mathcal{B}_2$ の記述であり、$\mathcal{A}$ は禁止された鍵生成クエリを行わないため、$\mathcal{B}_2$ も禁止された鍵生成クエリを行わないことに注意されたい。$\mathcal{B}_2$ と挑戦者の間で行われるゲームのチャレンジビットを β とする。$\beta = 0$ のとき、$\mathcal{B}_2$ は挑戦者から全ての $i \in [n]$ についてチャレンジ暗号文 $c^*_{i,1\oplus\alpha_i} \leftarrow \mathsf{Enc}(pp, (id_i, 1\oplus\alpha_i), m_i^*)$ を受け取る。よって、$\mathcal{B}_2$ は $\mathcal{A}$ に対して $\mathbf{Game}_1$ を完全にシミュレートしているため、$\mathcal{D}$ に入力される out' の分布は、$\mathbf{Game}_1$ での分布と完全に同一である。一方で、$\beta = 1$ のとき、$\mathcal{B}_2$ は挑戦者から全ての $i \in [n]$ についてチャレンジ暗号文 $c^*_{i,1\oplus\alpha_i} \leftarrow \mathsf{Enc}(pp, (id_i, 1\oplus\alpha_i), 1\oplus m_i^*)$ を受け取る。よって、$\mathcal{B}_2$ は $\mathcal{A}$ に対して $\mathbf{Game}_2$ を完全にシミュレートしているため、$\mathcal{D}$ に入力される out' の分布は、$\mathbf{Game}_2$ での分布と完全に同一である。したがって、$\mathsf{Adv}^{\mathsf{ind\text{-}id\text{-}cpa}}_{\Pi,\mathcal{B}_2}(\lambda) = 2\cdot|\Pr[\beta=\beta'] - \frac{1}{2}| = |\Pr[\beta'=1|\beta=0] - \Pr[\beta'=1|\beta=1]| = |\Pr[\mathbf{T}_1] - \Pr[\mathbf{T}_2]|$ が成り立つ。

□ **(補題 2)**

補題 3 $|\Pr[\mathbf{T}_2] - \Pr[\mathbf{T}_3]| \le \mathsf{Adv}^{\mathsf{s\text{-}sound}}_{\mathsf{PS},\mathcal{B}_3}(\lambda)$ を満たす PPT 攻撃者 $\mathcal{B}_3$ が存在する。

補題 3 の証明

各 $i \in \{2,3\}$ について、$\mathbf{Game}_i$ における事象 $\mathbf{Bad}_i$ を次のように定義する。

$\mathbf{Bad}_i$: $\mathcal{A}$ が $\mathbf{Game}_i$ にて $(\mathsf{Dec}(pp, sk_{id,0}, c_0) \neq \mathsf{Dec}(pp, sk_{id,1}, c_1)) \wedge (\mathsf{Verify}(crs, (pp, id, c_0, c_1), \pi) = 1)$ を満たす復号クエリ $(id, c = (c_0, c_1, \pi))$ を行う事象。(以下、このようなクエリを bad クエリと呼ぶ。)

$\mathbf{Game}_2$ は、$\mathbf{Bad}_2$ が起こらない限り、$\mathbf{Game}_3$ と同じゲームである。よって、$|\Pr[\mathbf{T}_2] - \Pr[\mathbf{T}_3]| \le \Pr[\mathbf{Bad}_2] = \Pr[\mathbf{Bad}_3]$ が成り立つ。以下では、Π' に対する攻撃者 $\mathcal{A}$ を用いて、$\Pr[\mathbf{Bad}_2] = \mathsf{Adv}^{\mathsf{s\text{-}sound}}_{\mathsf{PS},\mathcal{B}_3}(\lambda)$ となるような PS のシミュレーション健全性に対する PPT 攻撃者 $\mathcal{B}_3$ を構成する。

(1) $\mathcal{B}_3$ は挑戦者から crs を受け取り、$(pp, msk) \leftarrow \mathsf{Setup}(1^\lambda)$ を計算して、$PP := (pp, crs)$ とセットして、$\mathcal{A}$ に PP を送る。$\mathcal{B}_3$ は、自分の鍵生成リスト $\mathsf{L}^{\mathcal{B}_3} := \emptyset$ と鍵抽出リスト $\mathsf{L}^{\mathcal{B}_3}_{\mathsf{ext}} := \emptyset$ 用意する。

鍵生成クエリ $\mathcal{A}$ の鍵生成クエリ id に対して、$\mathcal{B}_3$ はまず $(id, SK'_{id} = (\perp, sk_{id,0}, sk_{id,1}, id)) \in \mathsf{L}^{\mathcal{B}_3}\backslash\mathsf{L}^{\mathcal{B}_3}_{\mathsf{ext}}$ かどうかをチェックする。$(id, SK'_{id}) \in \mathsf{L}^{\mathcal{B}_3}\backslash\mathsf{L}^{\mathcal{B}_3}_{\mathsf{ext}}$ の場合のみ、$\mathcal{B}_3$ は α を $\{0,1\}$ から一様ランダムにサンプルし、$SK'_{id} := (\alpha, sk_{id,0}, sk_{id,1}, id)$ と更新する。
その後、$\mathcal{B}_3$ は $(id, SK'_{id}) \in \mathsf{L}^{\mathcal{B}_3}$ かどうかをチェックする。$(id, SK_{id}) \in \mathsf{L}^{\mathcal{B}_3}$ である場合、$\mathcal{B}_3$ は既に生成された $SK'_{id} = (\alpha, sk_{id,0}, sk_{id,1}, id)$ を用いて、$SK_{id} := (\alpha, sk_{id,\alpha}, id)$ を $\mathcal{A}$ に返す。そうでないならば、$\mathcal{B}_3$ は α を $\{0,1\}$ から一様ランダムにサンプルし、$sk_{id,0} \leftarrow \mathsf{KG}(msk, (id, 0))$ と $sk_{id,1} \leftarrow \mathsf{KG}(msk, (id, 1))$ を計算する。そして、$\mathcal{B}_3$ は $\mathcal{A}$ に $SK_{id} := (\alpha, sk_{id,\alpha}, id)$ を返し、$SK'_{id} := (\alpha, sk_{id,0}, sk_{id,1}, id)$ とセットして、$\mathsf{L}^{\mathcal{B}_3}$ と $\mathsf{L}^{\mathcal{B}_3}_{\mathsf{ext}}$ に (id, SK'_{id}) を追加する。
復号クエリ $\mathcal{A}$ の復号クエリ (id, c) に対して、$\mathcal{B}_3$ はまず $(id, SK'_{id}) \in \mathsf{L}^{\mathcal{B}_3}$ かどうかをチェックする。$(id, SK'_{id}) \notin \mathsf{L}^{\mathcal{B}_3}$ の場合のみ、$\mathcal{B}_3$ はまず α を $\{0,1\}$ から一様ランダムにサンプルし、$sk_{id,0} \leftarrow \mathsf{KG}(msk, (id, 0))$ と $sk_{id,1} \leftarrow \mathsf{KG}(msk, (id, 1))$ を計算する。そして、$\mathcal{B}_3$ は $SK'_{id} := (\alpha, sk_{id,0}, sk_{id,1}, id)$ とセットして、$\mathsf{L}^{\mathcal{B}_3}$ に (id, SK'_{id}) を追加する。
その後、$\mathcal{B}_3$ は $c = (c_0, c_1, \pi)$ に分割して、既に生成された $SK'_{id} = (\alpha, sk_{id,0}, sk_{id,1}, id)$ を用いて、$(\mathsf{Dec}(pp, sk_{id,0}, c_0) \neq \mathsf{Dec}(pp, sk_{id,1}, c_1)) \wedge (\mathsf{Verify}(crs, (pp, id, c_0, c_1), \pi) = 1)$ が成り立つかどうかを検証する。もし成り立つならば、$\mathcal{B}_3$ は $x^* := (pp, id, c_0, c_1), \pi^* := \pi$ とセットして、挑戦者に (x^*, π^*) を送り、ゲームを強制終了する。成り立たないならば、$\mathcal{B}_3$ は $m \leftarrow \mathsf{Dec}'(PP, SK_{id}, c)$ を $\mathcal{A}$ に返し、ゲームを続行する。

(2) $\mathcal{B}_3$ は $\mathcal{A}$ から n 個のチャレンジ ID $(id_i)_{i\in[n]}$ と平文分布 Dist を受け取り、まずチャレンジ平文 $(m_i^*) \leftarrow \mathsf{Dist}$ をサンプルし、全てのチャレンジ ID $(id_i)_{i\in[n]}$ に対して $(id_i, SK_{id_i}) \in \mathsf{L}^{\mathcal{B}_3}\backslash\mathsf{L}^{\mathcal{B}_3}_{\mathsf{ext}}$ かどうかをチェックする。$(id_i, SK'_{id_i}) \in \mathsf{L}^{\mathcal{S}}\backslash\mathsf{L}^{\mathcal{B}_3}_{\mathsf{ext}}$ の場合のみ、$\mathcal{B}_3$ は α_i を $\{0,1\}$ から一様ランダムにサンプルし、$SK'_{id_i} := (\alpha_i, sk_{id_i,0}, sk_{id_i,1}, id_i)$ と更新する。
その後、$\mathcal{B}_3$ は全ての $i \in [n]$ に対して、$(id_i, SK'_{id_i}) \in \mathsf{L}^{\mathcal{B}_3}$ かどうかをチェックする。$(id_i, SK_{id_i}) \notin \mathsf{L}^{\mathcal{B}_3}$ の場合のみ、$\mathcal{B}_3$ は α を $\{0,1\}$ から一様ランダムにサンプルし、$sk_{id,0} \leftarrow \mathsf{KG}(msk, (id, 0))$ と $sk_{id,1} \leftarrow \mathsf{KG}(msk, (id, 1))$ を計算し、$SK'_{id} := (\alpha, sk_{id,0}, sk_{id,1}, id)$ とセットして、$\mathsf{L}^{\mathcal{B}_3}$ に (id, SK'_{id}) を追加する。その後、$\mathcal{B}_3$ は全ての $i \in [n]$ に対して、SK'_{id_i} 内の $\alpha_i \in \{0,1\}$ に基づいて、$c^*_{i,\alpha_i} \leftarrow \mathsf{Enc}(pp, (id_i, \alpha_i), m_i^*)$ と $c^*_{i,1\oplus\alpha_i} \leftarrow \mathsf{Enc}(pp, (id_i, 1\oplus\alpha_i), 1\oplus m_i^*)$ を計算して、$x_i^* := (pp, id_i, c^*_{i,0}, c^*_{i,1})$ とセットして、挑戦者にシミュレーションクエリ x_i^* を送る。そして、$\mathcal{B}_3$ は挑戦者から証明 π_i^* を受け取り、$c_i^* := (c^*_{i,0}, c^*_{i,1}, \pi_i^*)$ とセットする。最後に、$\mathcal{B}_3$ は $\mathcal{A}$ に $\mathbf{c}^* := (c_i^*)_{i\in[n]}$ を返す。$\mathcal{B}_3$ は $\mathcal{A}$ からの鍵生成クエリと復号クエリに対して、(1) のときと同様に答える。

(3) $\mathcal{B}_3$ は、$\mathcal{A}$ からインデックス集合 $\mathcal{J} \subset [n]$ を受け取り、$\mathsf{L}^{\mathcal{B}_3}$ 内の $(SK_{id_j})_{j\in\mathcal{J}}$ を用いて $\mathbf{SK}^*_{\mathcal{J}} := (SK_{id_j})_{j\in\mathcal{J}}$ とセットして、$\mathbf{SK}^*_{\mathcal{J}}$ を $\mathcal{A}$ に返す。$\mathcal{B}_3$ は $\mathcal{A}$ からの鍵生成クエリと復号クエリに対して、(1) のときと同様に答える。

(4) $\mathcal{B}_3$ は、$\mathcal{A}$ が out を送ってきたら、ゲームに勝利することを諦める。

上記の $\mathcal{B}_3$ の構成から、$\mathcal{B}_3$ が $\mathcal{A}$ に対して $\mathbf{Game}_2$ をシミュレートしていることは明らかである。ここで、$\mathcal{B}_3$ の勝利条件は、$((x,\pi) \notin \mathsf{L}_{\text{s-sound}}) \wedge (\mathsf{Verify}(crs,x,\pi)=1) \wedge (x \notin \mathcal{L}_{eq})$ を満たす命題と証明のペア (x,π) を出力することである。(ただし、$x=(pk_0,pk_1,c_0,c_1)$ であり、$\mathsf{L}_{\text{s-sound}}$ は $\mathcal{B}_3$ が挑戦者にシミュレーションクエリした x' とそれに対する証明 π' のペア (x',π') を保存したリストである。) もし $\mathcal{A}$ が bad クエリ $(id, c=(c_0,c_1,\pi))$ を行なった場合、$(\mathsf{Dec}(pp,sk_{id,0},c_0) \neq \mathsf{Dec}(pp,sk_{id,1},c_1)) \wedge (\mathsf{Verify}(crs,(pp,id,c_0,c_1),\pi)=1)$ が成り立つ。よってまず、Π の正当性により、$x \notin \mathcal{L}_{eq}$ が成り立つ。

さらに、$(x,\pi) \notin \mathsf{L}_{\text{s-sound}}$ が成り立つことがわかる。具体的には、$id \neq id_i$ の場合、$\mathcal{B}_3$ は挑戦者にシミュレーションクエリ x を行なっていないため、$(x,\pi) \notin \mathsf{L}_{\text{s-sound}}$ が成り立つ。$id=id_i$ の場合でも、$\mathcal{A}$ が行える復号クエリの条件より $(c^*_{i,0},c^*_{i,1},\pi^*_i)=c^*_i \neq c=(c_0,c_1,\pi)$ であるから、$(x,\pi) \notin \mathsf{L}_{\text{s-sound}}$ が成り立つ。

したがって、$\mathcal{A}$ が bad クエリを行なった場合、$\mathcal{B}_3$ は (x,π) を挑戦者に返すことによりシミュレーション健全性の勝利条件を満たすことができる。$\mathcal{B}_3$ は、ユーザ秘密鍵 $sk_{id,0}$ と $sk_{id,1}$ を両方とも所持しているため、事象 $\mathbf{Bad}_2$ を検出できることに注意されたい。以上の議論より、$\Pr[\mathbf{Bad}_2] = \mathsf{Adv}^{\text{s-sound}}_{\mathsf{PS},\mathcal{B}_3}(\lambda)$ が成り立ち、$|\Pr[\mathbf{T}_2]-\Pr[\mathbf{T}_3]| \leq \mathsf{Adv}^{\text{s-sound}}_{\mathsf{PS},\mathcal{B}_3}(\lambda)$ が成り立つことが示された。 □ **(補題 3)**

補題 4 $|\Pr[\mathbf{T}_3]-\Pr[\mathbf{T}_4]|=0$ が成り立つ。

補題 4 の証明

$\alpha'_i = \alpha_i \oplus m^*_i$ とおく。α_i は $\{0,1\}$ から一様ランダムに選ばれているため、α'_i もまた $\{0,1\}$ 上で一様ランダムに振る舞う。この事実より、$\mathbf{Game}_3$ と $\mathbf{Game}_4$ で $\mathcal{A}$ の view は全く同じであることがわかる。具体的には、$\mathbf{Game}_3$ では、全てのチャレンジ ID $(id_i)_{i\in[n]}$ について $\mathcal{A}$ に与えられるチャレンジ暗号文の要素 $c^*_{i,0},c^*_{i,1}$ はそれぞれ $c^*_{i,\alpha_i}=\mathsf{Enc}(pp,(id_i,\alpha_i),m^*_i), c^*_{i,1\oplus\alpha_i}=\mathsf{Enc}(pp,(id_i,1\oplus\alpha_i),1\oplus m^*_i)$ であり、$\mathcal{A}$ に与えられる秘密鍵 SK_{id_i} は $(\alpha_i, sk_{id_i,\alpha_i}, id_i)$ である。一方で、$\mathbf{Game}_4$ では、全てのチャレンジ ID $(id_i)_{i\in[n]}$ について $\mathcal{A}$ に与えられるチャレンジ暗号文の要素 $c^*_{i,0},c^*_{i,1}$ はそれぞれ $c^*_{i,\alpha'_i}=\mathsf{Enc}(pp,(id_i,\alpha'_i),m^*_i), c^*_{i,1\oplus\alpha'_i}=\mathsf{Enc}(pp,(id_i,1\oplus\alpha'_i),1\oplus m^*_i)$ であり、$\mathcal{A}$ に与えられる秘密鍵 SK_{id_i} は $(\alpha'_i, sk_{i,\alpha'_i}, id_i)$ である。よって、$|\Pr[\mathbf{T}_3]-\Pr[\mathbf{T}_4]|=0$ が成り立つ。 □ **(補題 4)**

補題 5 $|\Pr[\mathbf{T}_4]-\Pr[\mathbf{T}_5]|=0$ が成り立つ。

補題 5 の証明

$\mathbf{Game}_4$ で $\mathcal{A}$ が受け取る情報が、$\mathbf{Game}_5$ で得られる情報と全く同じであることを示す。まず第一に、$\mathbf{Game}_4$ で $\mathcal{A}$ に与えられている公開パラメータ $PP=(pp,crs)$ は、$\mathbf{Game}_5$ でも全く同じである。第二に、全てのチャレンジ ID $(id_i)_{i\in[n]}$ について $\mathbf{Game}_4$ のチャレンジ暗号文の要素 $c^*_{i,0}$ と $c^*_{i,1}$ は、$c^*_{i,\alpha_i\oplus m^*_i}=\mathsf{Enc}(pp,(id_i,\alpha_i\oplus m^*_i),m^*_i), c^*_{i,1\oplus(\alpha_i\oplus m^*_i)}=\mathsf{Enc}(pp,(id_i,1\oplus(\alpha_i\oplus m^*_i)),1\oplus m^*_i)$ である。これらのチャレンジ暗号文の要素は、チャレンジ平文 $m^*_i\in\{0,1\}$ の値に依存していないことが分かる。具体的には、$c^*_{i,0}$ も $c^*_{i,1}$ もチャレンジ平文 $m^*_i\in\{0,1\}$ の値に関係なく、$c^*_{i,\alpha_i}\leftarrow\mathsf{Enc}(pp,(id_i,\alpha_i),0), c^*_{i,1\oplus\alpha_i}\leftarrow\mathsf{Enc}(pp,(id_i,1\oplus\alpha_i),1)$ と計算されていることが分かる。これは $\mathbf{Game}_5$ によって計算されるチャレンジ暗号文と同じである。第三に、$\mathbf{Game}_4$ で $\mathcal{A}$ が鍵生成クエリと復号クエリから得ている情報は、$\mathbf{Game}_5$ で $\mathcal{A}$ が得るものと同じである。最後に、全てのチャレンジ ID $(id_i)_{i\in[n]}$ について $\mathbf{Game}_4$ で $\mathcal{A}$ が受け取る秘密鍵 $SK_{id_i}=(\alpha_i\oplus m^*_i, sk_{id_i,\alpha_i\oplus m^*_i}, id_i)$ は、$\mathbf{Game}_5$ で $\mathcal{A}$ が得ている秘密鍵と同じである。以上より、$|\Pr[\mathbf{T}_4]-\Pr[\mathbf{T}_5]|=0$ が成り立つ。 □ **(補題 5)**

上記の補題 1, 2, 3, 4, 5 を合わせることにより、

$$\begin{aligned}
&\mathsf{Adv}^{\text{rso-cca}}_{\Pi',\mathcal{A},\mathcal{S},\mathcal{D}}(\lambda)\\
&=|\Pr[\mathbf{T}_0]-\Pr[\mathbf{T}_5]|\\
&\leq\sum_{i=0}^{4}|\Pr[\mathbf{T}_i]-\Pr[\mathbf{T}_{i+1}]|\\
&\leq\mathsf{Adv}^{\text{zk}}_{\mathsf{PS},\mathcal{B}_1}(\lambda)+\mathsf{Adv}^{\text{ind-id-cpa}}_{\Pi,\mathcal{B}_2}(\lambda)+\mathsf{Adv}^{\text{s-sound}}_{\mathsf{PS},\mathcal{B}_3}(\lambda)
\end{aligned}$$

が成り立つ。

上記の議論において、$\mathcal{A}, n, \mathcal{D}$ は任意に選ばれている。したがって、上の式に Π の IND-ID-CPA 安全性と PS のシミュレーション健全性とゼロ知識性を合わせると、任意の n と PPT 攻撃者 $\mathcal{A}$ に対して PPT シミュレータ $\mathcal{S}$ が存在して、任意の PPT 識別者 $\mathcal{D}$ に対して、$\mathsf{Adv}^{\text{rso-cca}}_{\Pi',\mathcal{A},\mathcal{S},\mathcal{D}}(\lambda)=\mathsf{negl}(\lambda)$ が成り立つ。以上より、Π' は RSO-CCA 安全性を満たすことが示された。 □ **(定理 1)**

参考文献

[1] S. Agrawal, D. Boneh, and X. Boyen. Efficient Lattice (H)IBE in the Standard Model. *EUROCRYPT 2010*, pp. 553–572.

[2] M. Bellare, R. Dowsley, B. Waters, and S. Yilek. Standard Security Does Not Imply Security against Selective-Opening. *EUROCRYPT 2012*, pp. 645–662.

[3] M. Bellare, D. Hofheinz, and S. Yilek. Possibility and Impossibility Results for Encryption and Commitment Secure under Selective Opening. *EUROCRYPT 2009*, pp. 1–35.

[4] M. Bellare, B. Waters, and S. Yilek. Identity-Based Encryption Secure against Selective Opening Attack. *TCC 2011*, pp. 235–252.

[5] M. Bellare and S. Yilek. Encryption Schemes Secure under Selective Opening Attack. http://eprint.iacr.org/2009/101.

[6] D. Boneh and M. Franklin. Identity-based encryption from the Weil pairing. *CRYPTO 2001*, pp. 213–229.

[7] D. Boneh and X. Boyen. Secure Identity Based Encryption Without Random Oracles. *CRYPTO 2004*, pp. 443–459.

[8] J. Groth and A. Sahai. Efficient Non-interactive Proof Systems for Bilinear Groups. *EUROCRYPT 2008*, pp. 443–459.

[9] C. Hazay, A. Patra, and B. Warinschi. Selective Opening Security for Receivers. *ASIACRYPT 2015*, pp. 443–469.

[10] K. Hara, F. Kitagawa, T. Matsuda, G. Hanaoka, and K. Tanaka. Simulation-Based Receiver Selective Opening CCA Secure PKE from Standard Computational Assumptions. *SCN 2018*, pp. 140–159.

[11] Z. Huang, J. Lai, W. Chen, M. H. Au, Z. Peng, and J. Li. Simulation-based selective opening security for receivers under chosen-ciphertext attacks. *Designs, Codes and Cryptography.*, 87(6):1345–1371, 2019.

[12] D. Jia, X. Lu, and B. Li. Receiver Selective Opening Security from Indistinguishability Obfuscation. *INDOCRYPT 2016*, pp. 393–410.

[13] F. Kitagawa and K. Tanaka. Key Dependent Message Security and Receiver Selective Opening Security for Identity-Based Encryption. *PKC 2018*, pp. 32–61.

[14] M. Naor and M. Yung. Public-key Cryptosystems Provably Secure against Chosen Ciphertext Attacks. *STOC 1990*, pp. 427–437.

[15] J. Lai, R. H. Deng, S. Liu, J. Weng, Y. Zhao. Identity-Based Encryption Secure against Selective Opening Chosen-Ciphertext Attack. *EUROCRYPT 2014*, pp. 77–92.

[16] Y. Lindell. A Simpler Construction of CCA2-Secure Public-Key Encryption under General Assumptions. *EUROCRYPT 2003*, pp. 241–254.

[17] C. Peikert and S. Shiehian. Noninteractive Zero Knowledge for NP from (Plain) Learning with Errors. *CRYPTO 2019*, pp. 89–114.

[18] A. Sahai. Simulation-sound non-interactive zero knowledge. *Technical report, IBM RESEARCH REPORT RZ 3076, 2001.*

[19] A. Shamir. Identity-based cryptosystems and signature schemes. *CRYPTO 1984*, pp. 47–53.