

修士論文

適応的安全な  
属性ベース Signcryption  
に関する研究

**A Study on Adaptively Secure  
Attribute-Based Signcryption**

指導教員 松浦 幹太 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

48-126405 石坂 理人

平成28年2月4日提出



## 内容梗概

暗号技術は、科学技術の進歩が著しい現代のネットワーク環境において、安全性や利便性を高めるための機能を実現し、最も重要な情報技術の一つに数えられる。秘匿性や完全性を保障する機能を実現する公開鍵暗号や電子署名は最も基本的かつ最も重要な暗号技術の一つであるが、近年は様々な実用環境を想定したより応用的な“高機能公開鍵暗号技術”に関する研究が活発に行われている。本研究では、“属性情報”を利用した高機能公開鍵暗号技術である、属性ベース暗号、属性ベース署名、属性ベース Signcryption に着目する。

暗号文ポリシー型属性ベース暗号 (CP-ABE) は、平文がアクセス構造 (直感的には変数として属性を用い、演算子として AND や OR 等を用いる論理式) と関連付けられた上で暗号化され、そのアクセス構造を満たす様な属性集合を所持するユーザのみが属性集合に対応する秘密鍵を利用して暗号文を正しく復号して平文を入手できる暗号技術である。また、アクセス構造と秘密鍵が対応し、属性集合と暗号文 (平文) が対応する属性ベース暗号は、鍵ポリシー型属性ベース暗号 (KP-ABE) と呼ばれている。

署名ポリシー型属性ベース署名 (SP-ABS) によって、署名者はある文書  $M$  に関して自身が所持する属性集合を満たすようなアクセス構造を適切に指定した上で属性集合に対応する秘密鍵を用いて署名を作成し、署名検証者は署名検証を行う事でそのアクセス構造を満たす属性集合を所持する人物がその署名を文書  $M$  から作成した事実を確認できる。また、アクセス構造と秘密鍵が対応し、属性集合と署名が対応する属性ベース署名は、鍵ポリシー型属性ベース署名と呼ばれている。

暗号文ポリシー型属性ベース Signcryption (CP-ABSC) は、CP-ABE と SP-ABS の両機能を実現する暗号技術である。また、KP-ABE と KP-ABS の両機能を実現する暗号技術は、鍵ポリシー型属性ベース Signcryption (KP-ABSC) と呼ばれている。

本研究の主な成果は4つある。

第一の成果は、最も強い安全性であると言われている適応的述語モデルでの IND-CCA 安全性、適応的述語モデルでの sEUF-CMA 安全性、完全匿名性という三つの安全性を達成可能な CP-ABSC の一般的構成法である。過去にも最強の安全性を達成可能な CP-ABSC 方式の構成法は提案されているが、本研究の構成法には既存研究と比べると、仮定が弱い等の優れた性質を持った方式の構成がより容易になる等の様々な利点や意義があることを本稿では強調する。

第二の成果は、最も強い安全性であると言われている適応的属性モデルでの IND-CCA 安全性、適応的属性モデルでの sEUF-CMA 安全性、完全匿名性という三つの安全性を達成可能な KP-ABSC の一般的構成法である。CP-ABSC と同様、本研究の構成法には既存研究と比べると様々な利点や意義があることを本稿では強調する。

第三の成果は、CP-ABE を構成要素とした、暗号文ポリシー型属性ベース鍵カプセル化メカニズム (CP-ABKEM) の一般的構成である。前述の CP-ABSC の一般的構成においては、CP-ABKEM を構成要素としているので、本成果には大きな意義がある。

第四の成果は、KP-ABE を構成要素とした、鍵ポリシー型属性ベース鍵カプセル化メカニズム (KP-ABKEM) の一般的構成である。前述の KP-ABSC の一般的構成においては、KP-ABKEM を構成要素としているので、本成果には大きな意義がある。

# 目次

内容梗概	1
<b>1 序論</b>	<b>5</b>
1.1 属性ベース Signcryption	5
1.2 証明可能安全性	9
1.3 暗号技術の一般的構成	10
1.4 本研究の成果	11
1.5 本稿の構成	13
<b>2 準備</b>	<b>15</b>
2.1 暗号文ポリシー型属性ベース暗号	15
2.1.1 AP-IND-CCA 安全性	16
2.1.2 SP-IND-CCA 安全性	17
2.1.3 復号者アクセス構造開示性	18
2.2 鍵ポリシー型属性ベース暗号	19
2.2.1 AA-IND-CCA 安全性	19
2.2.2 SA-IND-CCA 安全性	21
2.2.3 復号者属性集合開示性	22
2.3 暗号文ポリシー型属性ベース鍵カプセル化メカニズム	22
2.3.1 AP-IND-CCA 安全性	23
2.3.2 SP-IND-CCA 安全性	24
2.3.3 復号者アクセス構造開示性	25
2.4 鍵ポリシー型属性ベース鍵カプセル化メカニズム	26
2.4.1 AA-IND-CCA 安全性	26
2.4.2 SA-IND-CCA 安全性	28
2.4.3 復号者属性集合開示性	29
2.5 データカプセル化メカニズム	29
2.5.1 IND-CCA 安全性	30
2.5.2 1対1対応性	30
2.6 署名ポリシー型属性ベース署名	31
2.6.1 AP-sEUFCMA 安全性	31
2.6.2 SP-sEUFCMA 安全性	33
2.6.3 完全匿名性	34
2.6.4 署名者アクセス構造衝突困難性	34
2.7 鍵ポリシー型属性ベース署名	35

2.7.1	AA-sEUF-CMA 安全性	36
2.7.2	SA-sEUF-CMA 安全性	37
2.7.3	完全匿名性	39
2.7.4	署名者属性集合衝突困難性	39
2.8	暗号文ポリシー型属性ベース Signcryption	40
2.8.1	AP-IND-CCA 安全性	41
2.8.2	SP-IND-CCA 安全性	43
2.8.3	AP-sEUF-CMA 安全性	45
2.8.4	SP-sEUF-CMA 安全性	47
2.8.5	完全匿名性	49
2.8.6	復号者アクセス構造開示性	49
2.9	鍵ポリシー型属性ベース Signcryption	50
2.9.1	AA-IND-CCA 安全性	50
2.9.2	SA-IND-CCA 安全性	52
2.9.3	AA-sEUF-CMA 安全性	54
2.9.4	SA-sEUF-CMA 安全性	55
2.9.5	完全匿名性	57
2.9.6	復号者属性集合開示性	57
<b>3</b>	<b>属性ベース Signcryption の関連研究</b>	<b>59</b>
<b>4</b>	<b>暗号文ポリシー型属性ベース鍵カプセル化メカニズムの一般的構成</b>	<b>63</b>
4.1	本章の概要	63
4.2	提案する一般的構成法	63
4.3	安全性証明	64
<b>5</b>	<b>暗号文ポリシー型属性ベース Signcryption の一般的構成</b>	<b>71</b>
5.1	本章の概要	71
5.2	提案する一般的構成法	71
5.3	安全性証明	80
<b>6</b>	<b>鍵ポリシー型属性ベース鍵カプセル化メカニズムの一般的構成</b>	<b>101</b>
6.1	本章の概要	101
6.2	提案する一般的構成法	101
6.3	安全性証明	102
<b>7</b>	<b>鍵ポリシー型属性ベース Signcryption の一般的構成</b>	<b>107</b>
7.1	本章の概要	107
7.2	提案する一般的構成法	107
7.3	安全性証明	107

<b>8 議論</b>	<b>124</b>
8.1 本研究の意義	124
8.1.1 鍵カプセル化メカニズムの一般的構成 (4章,6章) の意義	124
8.1.2 暗号文ポリシー型属性ベース Signcryption の一般的構成 (5章) の意義	125
8.1.3 鍵ポリシー型属性ベース Signcryption の一般的構成 (7章) の意義	129
8.2 本研究の課題	130
8.2.1 非 Combined-Setup 型属性ベース Signcryption について	130
8.2.2 SP-ABS(resp. KP-ABS) の署名者アクセス構造開示性 (resp. 署名者属性集合開示性) について	133
<b>9 結論</b>	<b>134</b>
謝辞	135
参考文献	136
発表文献	140

# Chapter 1 序論

## 1.1 属性ベース Signcryption

暗号技術は、科学技術の進歩が著しい現代のネットワーク環境において、安全性や利便性を高めるための機能を実現し、最も重要な情報技術の一つに数えられる。

情報を暗号化して送受信する際、送受信者が同一の秘密鍵を用いる共通鍵暗号を利用する場合、第三者に盗聴される可能性のある通信路において如何にして鍵を共有すればよいのか、といういわゆる“鍵配送問題”が生じる。この問題を解決する暗号技術が、公開鍵暗号 (Public Key Encryption, PKE) である。公開鍵暗号方式を利用する場合、送信者は受信者の公開鍵を用いて暗号化し、受信者は自身の秘密鍵を用いて復号を行う。従って、公開鍵暗号を利用する場合、鍵配送問題は生じない。

**鍵カプセル化メカニズムとデータカプセル化メカニズム** 公開鍵暗号は、共通鍵暗号の欠点であった鍵配送問題を解決する。しかし、公開鍵暗号は、共通鍵暗号と比べて、計算速度が圧倒的に遅い。そこで、公開鍵暗号の利点である鍵配送問題の解決、共通鍵暗号の利点である計算速度の速さ、それらの両方を実現するために考え出された暗号方式が、ハイブリッド暗号である。ハイブリッド暗号は、Shoup[44]により、KEM/DEM フレームワークという名前で形式化された。鍵カプセル化メカニズム (Key Encapsulation Mechanism, KEM) は、ハイブリッド暗号において公開鍵暗号方式で共通鍵を暗号化及び復号する方式を表す。そして、データカプセル化メカニズム (Data Encapsulation Mechanism, DEM) は、ハイブリッド暗号において共通鍵暗号方式で平文を暗号化及び復号する方式を表す。

**電子署名** 1976年に Diffie と Hellman[45] は、メッセージの完全性を保障する暗号技術として、電子署名の概念を提唱した。電子署名方式を利用する場合、署名者は自身の秘密鍵を用いてメッセージに関する署名を作成し、署名検証者は署名者の公開鍵を用いて、正しい署名であるかどうかの検証を行う。

**Signcryption** Zheng[39] は、公開鍵暗号と電子署名の両方の機能を実現し、かつ、公開鍵暗号と電子署名を単純に組み合わせて同機能を実現するよりも、計算コストや通信オーバーヘッドに関して、圧倒的に効率が良い暗号技術を提案し、Zheng は当該論文中これを“Signcryption”と命名した。

Signcryption が達成すべき安全性は、秘匿性と偽造不可能性 (完全性) であるが、いずれの安全性にも、内部者安全性と外部者安全性という安全性の分類が存在する。外部者安全性は、送信者及び受信者以外の第三者に対する安全性である。それに対し

て内部者安全性は、送信者または受信者のいずれかを攻撃者と仮定した上でのその攻撃者に対する安全性である。

具体的には、秘匿性の内部者安全性は、送信者を攻撃者として扱う。この安全性は現実においては以下のような意味を持つ。つまり、あるユーザ（送信者）の秘密鍵が漏洩した場合に、そのユーザ（送信者）が、漏洩前または漏洩後にその漏洩した秘密鍵を用いて生成した（する）暗号文に関して、秘匿性に関して内部者安全な **Signcryption** 方式は、当該暗号文に関する秘匿性を完全に保障する。しかし、秘匿性に関して内部者安全でない **Signcryption** 方式は、当該暗号文に関する秘匿性を保障しない。

また、偽造不可能性の内部者安全性においては、受信者を攻撃者として扱う。この安全性は、「否認不可能性」と関連が深い。否認不可能性は、ある送信者がある受信者宛にあるメッセージを作成した場合、その送信者はそのメッセージを作成した事実を否定できないことを保障する。偽造不可能性に関して内部者安全な **Signcryption** 方式は、否認不可能性を保障する。逆に内部者安全でない方式はそれを保障しない。

両安全性の定義より明らかに内部者安全性は外部者安全性よりも真に強い。よって、内部者安全性は外部者安全性よりも、強い（望ましい）安全性である。そして、Chibaら [31] は、秘匿性、偽造不可能性のいずれに関しても「多人数モデルでの内部攻撃者に対する安全性」という最も強い安全性を達成可能であり、かつ既存の構成法よりも弱い仮定に安全性の根拠を置く、**Signcryption** の一般的構成法を提案した。

最後に **Signcryption** に関して補足しておくべきなのは、本パラグラフの冒頭で述べたように、1997年のZheng[39]による **Signcryption** の最初の定義においては、**Signcryption** は“公開鍵暗号と電子署名の両機能を実現可能であり、かつ、公開鍵暗号と電子署名を単純に組み合わせると同機能を実現する場合よりも、圧倒的に効率が良い”プリミティブであった。しかし、近年は **Signcryption** の定義は曖昧化しており、単純に“公開鍵暗号と電子署名の両機能を実現可能”であれば、それを **Signcryption** と呼ぶ向きもある。Chibaら [31] の **Signcryption** はまさに後者の意味での **Signcryption** である。

公開鍵暗号や電子署名等は最も基本的かつ最も重要な暗号技術であるが、近年は様々な実用環境を想定したより応用的な“高機能公開鍵暗号技術”に関する研究が活発に行われている。そして、“属性情報”を利用した、属性ベース暗号、属性ベース鍵カプセル化メカニズム、属性ベース署名、属性ベース **Signcryption** は、そのような高機能公開鍵暗号技術に分類される暗号技術である。

**属性ベース暗号** 属性ベース暗号 (Attribute-Based Encryption, ABE) は、属性情報を利用して暗号文の復号権限を柔軟に変更できることを特徴とした暗号技術である。ABEは機能の違いにより、暗号文ポリシー型と鍵ポリシー型に分類される。

暗号文ポリシー型属性ベース暗号 (Ciphertext-Policy Attribute-Based Encryption, CP-ABE) を用いる場合、前提としてシステムユーザは全員、個人を特徴づける情報である属性を割り振られているとする。そして、メッセージを暗号化して送る者は、暗号化時に“アクセス構造”と呼ばれるアクセス権限（復号権限）を設定する。アクセス構造は直感的には、“変数として属性を用い、演算子として AND や OR 等を用いて表される論理式”である。例えば、暗号文の作成者は、「“経理部” AND (“部長” OR

“課長”）」というアクセス構造を設定して暗号化することで、「“経理部”の“部長”」と「“経理部”の“課長”」だけが各々自身の秘密鍵を用いて正しく復号できる暗号文を作成することができる。要するに、CP-ABEは、暗号文とポリシー（アクセス構造）、秘密鍵と属性集合、がそれぞれ対応づけられる ABE である。それに対して、鍵ポリシー型属性ベース暗号 (Key-Policy Attribute-Based Encryption, KP-ABE) は、秘密鍵とポリシー（アクセス構造）、暗号文と属性集合、がそれぞれ対応づけられる ABE である。

CP-ABE の安全性は秘匿性（識別不可能性）である。そして、CP-ABE の秘匿性（識別不可能性）の安全性定義は、公開鍵暗号と同じ「“選択平文攻撃に対する安全性 (CPA)” または “適応的選択暗号文攻撃に対する安全性 (CCA)”」という分類と、「“選択的述語安全性 (Selective Predicate, SP)” または “適応的述語安全性 (Adaptive Predicate, AP)”」という分類が存在する。ちなみに、Yamada ら [40] により、CPA 安全な ABE を CCA 安全な ABE へ変換する比較的弱い仮定に基づく手法が提案されているため、前者の分類にはあまり安全性の差がないと言える。従って、CP-ABE の秘匿性（偽造不可能性）の強さを考える上で重要なのは後者の分類である。そして、適応的述語安全性が選択的述語安全性よりも真に強い安全性であることは定義 (2.1.1 項, 2.1.2 項を参照) より自明であり、適応的述語モデルでの IND-CCA 安全性 (AP-IND-CCA 安全性) (2.1.1 項) が CP-ABE の秘匿性（識別不可能性）の最も強い安全性であるとされる。以上の議論は、KP-ABE に関しても同様に成り立つ。KP-ABE の場合、適応的属性モデルでの IND-CCA 安全性 (2.2.1 項) が最強の安全性であるとされる。

属性ベース暗号は、Sahai ら [41] による “Fuzzy Identity-Based Encryption” という方式の提案を起源とする。これは、暗号文（平文）に任意の属性集合  $X$  を関連付け、秘密鍵に任意の属性集合  $Y$  を関連付け、属性集合  $X$  と  $Y$  の中の属性が一定個数以上重複すれば、その暗号文はその秘密鍵で正しく復号できるという機能であった。その後、Goyal ら [42] により、はじめての KP-ABE 方式が提案された。更にその後、Bethencourt ら [43] により、はじめての CP-ABE 方式が提案された。現在は様々な研究動機の下で ABE 方式の提案が行われており、安全性として最強の安全性（適応的モデルでの IND-CCA 安全性）を達成することを目的とした方式 [22, 23, 20, 21, 10, 11], “NMA (Non-Monotone Access Structure)” や “Large Universe”, “Unbounded” などの優れた性質（各性質の説明は 8.1.2 項を参照。）を持つ方式 [16, 22, 23, 18, 19, 10, 11], 弱い仮定に基づく方式 [20, 21], 高い効率を達成できる方式 [13, 10, 11], など様々な ABE 方式が提案されている。

**属性ベース鍵カプセル化メカニズム** 属性ベース鍵カプセル化メカニズム (Attribute-Based Key Encapsulation Mechanism, ABKEM) は、鍵カプセル化メカニズムの属性ベース版である。属性ベース暗号と同様、暗号文ポリシー型鍵カプセル化メカニズム (Ciphertext-Policy Attribute-Based Key Encapsulation Mechanism, CP-ABKEM) と、鍵ポリシー型鍵カプセル化メカニズム (Key-Policy Attribute-Based Encapsulation Mechanism, KP-ABKEM) の二種類が存在する。ABKEM については、ABE と比べると、研究の活発度合いが格段に低い。これまでに提案された ABKEM 方式は、選択的述語モデルでの選択暗号文攻撃者に対する一方向性安全性という非常に弱い安全性を達成可能な

CP-ABKEM 方式 [34], 安全性の仮定としてランダムオラクルモデルという強い仮定を用いている CP-ABKEM 方式 [36, 37] などが存在する。

**属性ベース署名** 属性ベース署名 (Attribute-Based Signature, ABS) は, 属性を用いて匿名的に署名機能を実現する技術である。ABS は署名ポリシー型と鍵ポリシー型の二種類が存在する。

署名ポリシー型属性ベース署名 (Signature-Policy Attribute-Based Signature) を用いる場合, 署名者はある文書  $M$  に関して自身が所持する属性集合が満たすようなアクセス構造を適切に指定した上で, 属性集合が関連付けられた秘密鍵を用いて署名を作成する。その後, 署名検証者はその署名に関して署名検証を行う事で, そのアクセス構造を満たすような属性集合を所持する人物がその署名を文書  $M$  から作成した事実を確認できる。SP-ABS は, 署名とポリシー (アクセス構造), 秘密鍵と属性集合, がそれぞれ対応づけられる ABS である。それに対して, 鍵ポリシー型属性ベース署名 (Key-Policy Attribute-Based Signature, KP-ABS) は, 秘密鍵とポリシー, 署名と属性集合, がそれぞれ対応づけられる ABS である。

ABS の安全性は, 完全性 (偽造不可能性), 完全匿名性である。SP-ABS (resp. KP-ABS) の完全匿名性は, 署名から実際の署名作成者の属性 (resp. ポリシー) に関する情報が一切漏れないことを保障する安全性である。また, SP-ABS の完全性 (偽造不可能性) の安全性定義は, 電子署名と同様の「“適応的選択文書攻撃に対する弱存在的偽造不可能性 (wEUF-CMA)” または “適応的選択文書攻撃に対する強存在的偽造不可能性 (sEUF-CMA)”」という分類と, 「“選択的述語安全性” または “適応的述語安全性”」という分類が存在する。sEUF-CMA が wEUF-CMA よりも, 真に強い安全性であることは定義から自明である。同様に, 適応的述語安全性が選択的述語安全性よりも, 真に強い安全性であることは定義 (2.6.1 項, 2.6.2 項を参照) より自明であり, 適応的述語モデルでの sEUF-CMA (AP-sEUF-CMA 安全性) (2.6.1 項を参照) が, SP-ABS の完全性 (偽造不可能性) の最も強い安全性であるとされる。以上の議論は, KP-ABS に関しても同様に成り立つ。KP-ABS の場合, 適応的属性モデルでの sEUF-CMA (AA-sEUF-CMA 安全性) (2.7.1 項を参照) が最強の安全性であるとされる。

属性ベース署名は, Maji ら [26, 27] による SP-ABS 方式の提案に起源がある。これまでに, 安全性として最強の安全性を達成可能である方式 [26, 27, 28, 29, 9], “NMA (Non-Monotone Access Structure)” や “Large Universe”, “Unbounded” などの優れた性質 (各性質の説明は 8.1.2 項を参照。) を持つ方式 [26, 27, 28, 29, 9], 弱い仮定に基づく方式 [26, 27, 28, 29] など, いくつかの ABS 方式が提案されている。

**属性ベース Signcryption** 属性ベース暗号では, 完全性が保障されないため, 受信者 (復号者) はその暗号文の本当の送信者 (作成者) がどのような属性を持つ人物であるかに確信が持てない。属性ベース署名では, メッセージの秘匿性が保障されない。これらの問題を解決する暗号技術が属性ベース Signcryption である。属性ベース Signcryption

は、属性ベース暗号と属性ベース署名の両機能を実現する暗号技術である<sup>1</sup>。ABSCは、暗号文ポリシー型と鍵ポリシー型の二種類が存在する。

暗号文ポリシー型属性ベース Signcryption(Ciphertext-Policy Attribute-Based Signcryption, CP-ABSC)は、CP-ABEとSP-ABSの両機能を同時に実現する暗号技術である。この技術によって、個人が属性によって識別・管理されるような環境において、通信データの秘匿性と完全性、完全匿名性の安全性が同時に保障される。具体的な応用例としては、クラウド上でのデータ共有サービス[32]、WBAN(Wireless Body Area Network)を利用した医療支援サービス[33]等が存在する。また、KP-ABEとKP-ABSの両機能を実現する暗号技術は、鍵ポリシー型属性ベース Signcryption(Key-Policy Attribute-Based Signcryption, KP-ABSC)と呼ばれている。

ABSCの安全性は、秘匿性(識別不可能性)、完全性(偽造不可能性)、完全匿名性である。CP-ABSCの秘匿性(識別不可能性)の最強の安全性は、適応的述語モデルでのIND-CCA安全性(AP-IND-CCA安全性)(2.8.1項を参照)である。また、CP-ABSCの偽造不可能性の最強の安全性は、適応的述語モデルでのsEUF-CMA安全性(AP-sEUF-CMA安全性)(2.8.3項を参照)である。ここで、ABSCの内部者安全性と外部者安全性について論じる。実は、既存研究で、ABSCの内部者安全性と外部者安全性について、明確な定義は行われていない。しかし、従来よりCP-ABSCの安全性として標準的に用いられており、本項でも用いている、AP-IND-CCA安全性及びAP-sEUF-CMA安全性は、実際には“内部者安全性”である(詳細は2.8.1項、2.8.3項を参照)。同様に、KP-ABSCの最強の安全性である、AA-IND-CCA安全性及びAA-sEUF-CMA安全性は、いずれも、“内部者安全性”である(詳細は2.9.1節、2.9.3節を参照)。

従って、適応的安全性、完全匿名性という最強の安全性を達成可能なABSC方式の構成法を考える上では、攻撃者は実際には“内部攻撃者”であることを考慮した上で適切な対策や構成法を考える必要がある。

## 1.2 証明可能安全性

あらゆる暗号技術は、現在知られている数学的に解くことが困難な問題に基づくなどして、証明可能安全性を持つことが望ましい。証明可能安全性とは、暗号の安全性を形式的に定義した上で、数学的に解くことが困難とされている問題について言及し、その問題を解くことができないという仮定を利用し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。ある暗号技術の安全性の証明がないことは、必ずしも安全ではないということに直接意味するわけではない。しかし、客観的な安全性の

<sup>1</sup>Signcryptionと同様、属性ベース Signcryptionの定義に関して、様々な解釈が存在することは事実である。第一に、“ABEとABSの両機能が実現可能な”暗号技術をABSCとする解釈。第二に、Signcryptionの本来のZheng[39]の定義に倣い、“ABEとABSの両機能を実現し、かつ両暗号技術を単純に組み合わせると同機能を実現するよりも、圧倒的に効率が良い”暗号技術をABSCとする解釈。第三に、[4][5][9]が明示的にそのように書いている訳ではないが、これらの論文内で用いられている“ABEとABSの両機能を実現し、かつ“Combined Setup型”である”暗号技術をABSCとする解釈。本稿のABSCはこの中の一番目の解釈によるABSCに含まれる。

議論を行うために、新たな暗号方式を提案する場合などでは、安全性の証明をつけることがほとんどである。

一概に証明可能安全性を持つ暗号技術といっても、どの安全性をもって安全とすればよいのかについては、暗号の設計者と利用者間に理解の溝があるのが事実である。実際、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は様々で、証明可能安全性を示すにはそれらの形式的な定義を行う必要がある。根拠となる問題は、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。

安全性の定義は“現実には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。すなわち，“安全性証明をしたい暗号方式の安全性を無視できない確率で破る確率多項式時間アルゴリズムが存在するならば、そのアルゴリズムを利用して、現実には知られている困難な問題を解くことができる”という事示す。確率多項式時間アルゴリズムは、現実には存在するアルゴリズムの能力を表しており、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。

また、本稿では安全性の定義を行う(2章を参照)にあたり、攻撃者と攻撃者を試すチャレンジャーによるゲームによって安全性を定式化している(新たな暗号方式を提案する際、安全性の定義を行うにはこのようなゲームを利用するが多い)。安全性の証明を行うにあたり、攻撃者にとってはこのゲームにおけるやりとりをしていることと変わりがなく、見分けがつかないようにできる事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使用する。

### 1.3 暗号技術の一般的構成

特定の安全性を満たす特定の暗号技術の一つまたは複数、ブラックボックス的に用いて、特定の安全性を満たす特定の暗号技術を構成することを、暗号技術の一般的構成と言う。

一般的構成はその安全性の根拠を、構成要素の安全性に置く。そのため、一般的構成の安全性証明で用いられる定理は、「(構成要素が)XXX 安全ならば、(一般的構成は)YYY 安全である」と書かれることが多い。実際にこの定理を証明する際には、この対偶を証明するという方法が採られる。つまり、一般的構成の YYY 安全性ゲームに勝利する事ができる攻撃者が存在するならば、その攻撃者をサブルーチンとして利用して、構成要素の XXX ゲームに勝利することができるアルゴリズム(本研究では“シミュレータ”と呼んでいる)を作ることができることを示す。これが真であれば対偶も真であるから、定理が成立するという論理である。

特定の暗号技術に関して一般的構成法を示すことには、様々な利点があるとされている。

利点の一つは、暗号技術の危殆化のリスクが減ることである。一般的構成でない具体的な構成の場合、その多くが、計算量理論的安全性によって安全性を保障されており、かつ安全性の根拠を数論等に基づく具体的な困難性仮定に置いている。数論等の具体的な困難性仮定は、常に“具体的な攻撃法が発見されて成立しないことが証明される”リスクが付きまとう。安全性の根拠に置いていた仮定が成り立たないことが証明されれば、当然元の具体的な構成も安全でないことになる。一般的構成の場合、安全性の根拠を構成要素の安全性に置いており、特定の具体的な困難性仮定に依存しない。ある困難性仮定が成り立たないことが証明されたとしても、成立することが強く信じられているそれとは別の仮定に安全性の根拠を置くような方式を構成要素として採用することで新たに安全な方式を構成できる。従って、一般的構成は具体的な構成と比べて、危殆化のリスクが小さいという利点がある。

大半の暗号技術が情報理論的安全性ではなく、計算量理論的安全性に基づいている現代の暗号理論の分野においては、特定の暗号技術に関してその一般的構成法を示すことは、非常に有意義な課題と考えられている。

## 1.4 本研究の成果

本研究の主な成果は4つある。

- 暗号文ポリシー型属性ベース鍵カプセル化メカニズム (CP-ABKEM) の一般的構成とその安全性証明 (4章)
- 暗号文ポリシー型属性ベース Signcryption(CP-ABSC) の一般的構成とその安全性証明 (5章)
- 鍵ポリシー型属性ベース鍵カプセル化メカニズム (KP-ABKEM) の一般的構成とその安全性証明 (6章)
- 鍵ポリシー型属性ベース Signcryption(KP-ABSC) の一般的構成とその安全性証明 (7章)

各成果の概要を以下で述べる。また、成果1と成果2のイメージ図を図1.1に示し、成果3と成果4のイメージ図を図1.2に示す。

**CP-ABKEMの一般的構成** 暗号文ポリシー型属性ベース暗号(CP-ABE)を構成要素とした、CP-ABKEMの一般的構成法を提案する。そして、安全性証明を行い、CP-ABEがAP-IND-CCA安全かつ復号者アクセス構造開示的であれば、提案したCP-ABKEMの一般的構成法はAP-IND-CCA安全性を達成可能であり、かつ復号者アクセス構造開示性を満たすことを証明する。

本成果の意義・貢献については、8.1.1項で詳述する。

**CP-ABSC の一般的構成** 暗号文ポリシー型属性ベース鍵カプセル化メカニズム (CP-ABKEM), 署名ポリシー型属性ベース署名 (SP-ABS), データカプセル化メカニズム (DEM) を構成要素とし, CP-ABSC の安全性の中で最強の安全性を達成可能な CP-ABSC の一般的構成法を提案する. 具体的には, CP-ABKEM が AP-IND-CCA 安全かつ復号者アクセス構造開示的, かつ SP-ABS が AP-sEUF-CMA 安全かつ完全匿名かつ署名者アクセス構造衝突困難, かつ DEM が IND-CCA 安全かつ一対一対応であれば, 提案した CP-ABSC の一般的構成法は, AP-IND-CCA 安全性, AP-sEUF-CMA 安全性, 完全匿名性という最強の安全性を達成可能であることを, 安全性証明を行い証明した.

本成果の意義・貢献については, 8.1.2 項で詳述する.

**KP-ABKEM の一般的構成** 鍵ポリシー型属性ベース暗号 (KP-ABE) を構成要素とした, KP-ABKEM の一般的構成法を提案する. そして, 安全性証明を行い, KP-ABE が AA-IND-CCA 安全かつ復号者属性集合開示的であれば, 提案した KP-ABKEM の一般的構成法は AA-IND-CCA 安全性を達成可能であり, かつ復号者属性集合開示性を満たすことを証明する.

本成果の意義・貢献については, 8.1.1 項で詳述する.

**KP-ABSC の一般的構成** 鍵ポリシー型属性ベース鍵カプセル化メカニズム (KP-ABKEM), 鍵ポリシー型属性ベース署名 (KP-ABS), データカプセル化メカニズム (DEM) を構成要素とし, KP-ABSC の安全性の中で最強の安全性を達成可能な KP-ABSC の一般的構成法を提案する. 具体的には, KP-ABKEM が AA-IND-CCA 安全かつ復号者属性集合開示的, かつ KP-ABS が AA-sEUF-CMA 安全かつ完全匿名かつ署名者属性集合衝突困難, かつ DEM が IND-CCA 安全かつ一対一対応であれば, 提案した KP-ABSC の一般的構成法は, AA-IND-CCA 安全性, AA-sEUF-CMA 安全性, 完全匿名性という最強の安全性を達成可能であることを, 安全性証明を行い証明した.

本成果の意義・貢献については, 8.1.3 項で詳述する.

なお, 本研究ではいくつかの暗号技術に関して, 新たな性質を独自に定義している. それは, “復号者アクセス構造開示性”, “署名者アクセス構造衝突困難性”, “復号者属性集合開示性”, “署名者属性集合衝突困難性” である. いずれの性質に関しても, 厳密な定義は 2 章で説明する. 復号者アクセス構造開示性と復号者属性集合開示性は, 2 章で説明するように, 既存方式の多くがこの性質を満たすことが自明な極めて自然な性質である. 署名者アクセス構造衝突困難性と署名者属性集合衝突困難性は, 既存方式がこの性質を満たすことの証明はまだできていないが, 直感的には既存方式が当該性質を満たす可能性は高く, 当該性質は自然な性質であると考えられる. 当該性質が自然な性質であると考えられる根拠等については 8.2.2 項で説明する.

また, 本研究で提案する CP-ABSC と KP-ABSC の一般的構成法は, いずれも非 “Combined Setup” 型である. 非 Combined Setup 型の ABSC にはユーザが管理すべき鍵の個数が増えるという欠点があるが, Combined Setup 型の ABSC と比べて利点もある. これに関する詳しい説明は, 本研究の成果を参考に Combined Setup 型の ABSC の一

般的構成を実現することは可能であるかどうかという疑問に対する回答と共に，8.2.1項で行う。

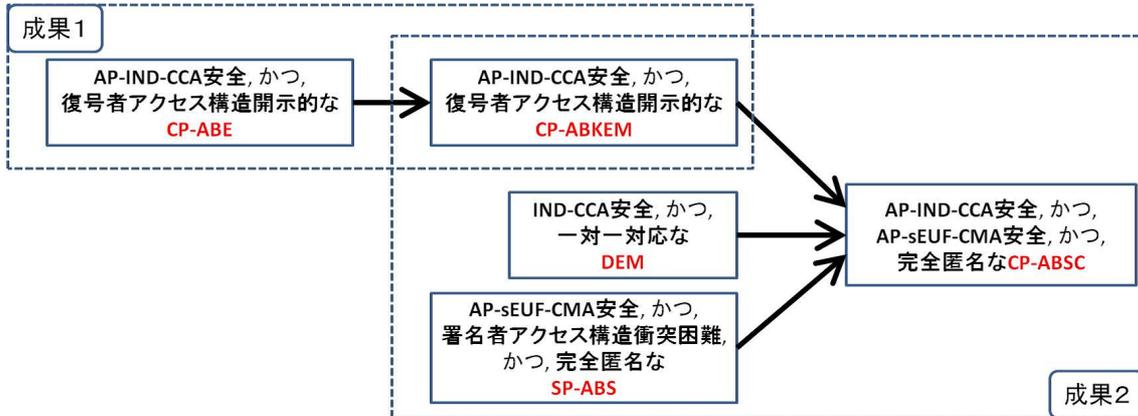


図 1.1: 成果 1 と成果 2 のイメージ

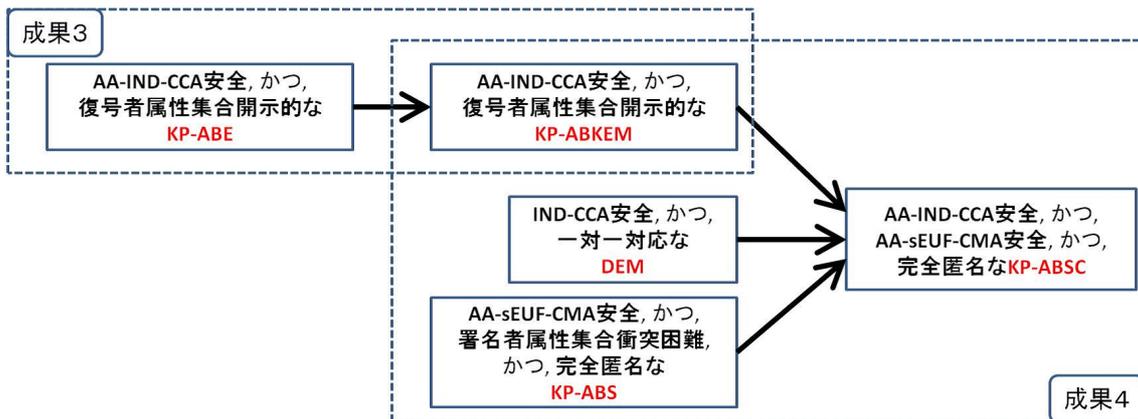


図 1.2: 成果 3 と成果 4 のイメージ

## 1.5 本稿の構成

次章以降の本稿の構成は以下の通りである．2章では，本研究で用いる全暗号技術のシンタックス(アルゴリズム)，安全性，性質等の諸定義を説明する．3章では，属性ベース **Signcryption** の関連研究を説明する．4章では，暗号文ポリシー型属性ベース鍵カプセル化メカニズムの一般的構成とその安全性証明を行う．5章では，暗号文ポリシー型属性ベース **Signcryption** の一般的構成とその安全性証明を行う．6章では，鍵ポリシー型属性ベース鍵カプセル化メカニズムの一般的構成とその安全性証明を行う．7章では，鍵ポリシー型属性ベース **Signcryption** の一般的構成とその安全性証明を行う．

8章では、議論を行う。具体的には、本研究の成果にはどのような意義があるか、また非 **Combined Setup** 型の属性ベース **Signcryption** にはどのような特徴(欠点等)があるか、**Combined Setup** 型の属性ベース **Signcryption** の一般的構成は可能であるか、また既存の **SP-ABS** 方式が署名者アクセス構造開示性と本研究で独自に命名しかつ定義した性質を満たさない場合にどのような問題が生じる可能性があるか、に関して議論を行う。9章では、結論を示す。

なお、5章の成果に関しては、査読なし国内会議 **SCIS2016**(発表文献 [i]) で発表した。

## Chapter 2 準備

**本稿で用いる記号の定義** 本論文内で使用する自明でない記号・数式・単語・表現等の意味の説明を行う.  $\forall x$ は「全ての  $x$  (について)」という意味を表す.  $x \leftarrow y$ はアルゴリズム (関数)  $y$ を実行した結果を変数  $x$ とする (代入する) 操作を表す.  $x \stackrel{\mathcal{U}}{\leftarrow} y$ は集合  $y$ からランダムに要素を取り出し変数  $x$ とする (代入する) 処理を表す.  $x := y$ は変数  $x$ を変数 (定数)  $y$ の値を持つものとして定義する操作を表す.  $x(\text{s.t.}R)$ は「条件  $R$ を満たすような  $x$ 」を表す.  $x||y$ は変数 (定数)  $x, y$ を連結する操作を表す. 「(ある関数が) 無視できる (ほど小さい)」という表現を用いる場合, 「セキュリティパラメータ  $k$ に関して」という意味を暗に含む.  $\mathcal{M}$ は平文空間,  $\mathcal{K}$ は鍵空間を表す.  $\mathcal{U}$ は属性の全体集合,  $\mathbb{A}$ はアクセス構造,  $S$ は属性集合を表す.  $\mathbb{A}_d$ は復号者アクセス構造,  $\mathbb{A}_s$ は署名者アクセス構造,  $S_d$ は復号者属性集合<sup>1</sup>,  $S_s$ は署名者属性集合<sup>2</sup>を表す. PPTAは確率的多項式時間アルゴリズム (Probabilistic Polynomial Time Algorithm) の略である.

以下に本稿で用いるアクセス構造の定義を示す.

**定義 2.1** (アクセス構造 (Access Structure)[6, 4, 5]).

アクセス構造  $\mathbb{A}$ は, 属性の全体集合  $\mathcal{U}$ のべき集合から空集合を除いた集合族の部分集合族である. つまり,  $\mathbb{A} \subseteq (2^{\mathcal{U}} - \{\emptyset\})$ .

### 2.1 暗号文ポリシー型属性ベース暗号

暗号文ポリシー型属性ベース暗号 (Ciphertext-Policy Attribute-Based Encryption, 以降略称として“CP-ABE”と表記することもある)は, 平文を属性情報を使って表現されるポリシーと呼ばれる復号条件を関連付けた上で暗号化し, そのポリシーを満たす様な属性集合を持つ人物だけが, 自身の秘密鍵を使って暗号文を復号し平文を入手する事ができる高機能公開鍵暗号である. CP-ABEは, 暗号文 (平文) とポリシー (復号者アクセス構造), 秘密鍵と属性集合がそれぞれ対応する.

CP-ABE方式は以下の4つのアルゴリズムから構成される.

$\text{CE.Setup}(1^k, \mathcal{U}) \rightarrow (\text{PK}, \text{MK})$

セットアップアルゴリズム  $\text{CE.Setup}$ は, セキュリティパラメータ  $k$ , 属性の全体集合  $\mathcal{U}$ を入力とし, システム公開鍵  $\text{PK}$  とマスター秘密鍵  $\text{MK}$ を出力する.

<sup>1</sup>“復号者が所持する属性集合”ではない. KP-ABE, KP-ABKEM, KP-ABSCにおいて, 暗号文 (サインクリプトテキスト)に関連付けられる属性集合である.

<sup>2</sup>“署名者が所持する属性集合”ではない. KP-ABS, KP-ABSCにおいて, 暗号文 (サインクリプトテキスト)に関連付けられる属性集合である.

$\text{CE.KeyGen}(\text{PK}, \text{MK}, S_r) \rightarrow \text{SK}_r$

鍵生成アルゴリズム  $\text{CE.KeyGen}$  は、公開鍵  $\text{PK}$ 、マスター秘密鍵  $\text{MK}$ 、受信者用属性集合  $S_r \in (2^{\mathcal{U}} - \{\phi\})$  を入力とし、受信者用秘密鍵  $\text{SK}_r$  を出力する。

$\text{CE.Enc}(\text{PK}, m, \mathbb{A}_d) \rightarrow C$

暗号化アルゴリズム  $\text{CE.Enc}$  は、システム公開鍵  $\text{PK}$ 、平文  $m$ 、復号者アクセス構造  $\mathbb{A}_d$  を入力とし、暗号文  $C$  を出力する。

$\text{Dec}(\text{PK}, C, \text{SK}_r) \rightarrow m / \perp$

復号アルゴリズム  $\text{CE.Dec}$  は、システム公開鍵  $\text{PK}$ 、暗号文  $C$ 、受信者用秘密鍵  $\text{SK}_r$  を入力とし、平文  $m \in \mathcal{M}$  または“復号不可”を表す特別な記号  $\perp$  を出力する。

**正当性 (Correctness)** CP-ABE 方式は正当であることが求められる。CP-ABE 方式が正当であるとは、全ての  $k$ 、全ての  $\mathcal{U}$ 、全ての  $(\text{PK}, \text{MK}) \leftarrow \text{CE.Setup}(1^k, \mathcal{U})$ 、全ての  $S_r \in (2^{\mathcal{U}} - \{\phi\})$ 、全ての  $\text{SK}_r \leftarrow \text{CE.KeyGen}(\text{PK}, \text{MK}, S_r)$ 、 $S_r \in \mathbb{A}_d$  を満たす全ての  $\mathbb{A}_d$ 、全ての  $C \leftarrow \text{CE.Enc}(\text{PK}, m, \mathbb{A}_d)$  に対して、以下の等式が成立する場合を言う。

$$\Pr[\text{CE.Dec}(\text{PK}, C, \text{SK}_r) = m] = 1 \quad (2.1)$$

## 2.1.1 AP-IND-CCA 安全性

適応的述語モデルにおける適応的選択暗号文攻撃に対する暗号文識別不可能性 (Ciphertext-Indistinguishability under Adaptively Chosen Ciphertext Attack in the Adaptive Predicate Model, AP-IND-CCA) は、CP-ABE 方式が満たすべき秘匿性に関する安全性の中で、最も強い安全性と考えられている。CP-ABE 方式  $\Pi_{\text{CE}}$  に関する AP-IND-CCA 安全性は、攻撃者  $\mathcal{A}$  と挑戦者  $\mathcal{CH}$  との間で行われる以下の AP-IND-CCA 安全性ゲームによって定義する。

**Setup Phase:**  $\mathcal{CH}$  は  $\text{CE.Setup}(1^k, \mathcal{U}) \rightarrow (\text{PK}, \text{MK})$  を実行し、システム公開鍵  $\text{PK}$  を  $\mathcal{A}$  に渡す。

**Query Phase 1:**  $\mathcal{A}$  は以下の各オラクルに対してクエリを任意回数発行することができる。

**秘密鍵生成:**  $\mathcal{A}$  は属性集合  $S_r \in (2^{\mathcal{U}} - \{\phi\})$  をクエリする。但し、Challenge Phase で選択するターゲット復号者アクセス構造  $\mathbb{A}_d^*$  に対して、 $S_r \in \mathbb{A}_d^*$  を満たす  $S_r$  のクエリは禁止とする。 $\mathcal{CH}$  は  $\text{CE.KeyGen}(\text{PK}, \text{MK}, S_r) \rightarrow \text{SK}_r$  を実行し  $\text{SK}_r$  を  $\mathcal{A}$  へ返す。

**復号:**  $\mathcal{A}$  は暗号文  $C$ 、属性集合  $S_r \in (2^{\mathcal{U}} - \{\phi\})$  をクエリする。 $\mathcal{CH}$  は  $\text{CE.KeyGen}(\text{PK}, \text{MK}, S_r) \rightarrow \text{SK}_r$ 、 $\text{CE.Dec}(\text{PK}, C, \text{SK}_r) \rightarrow m / \perp$  を実

行し, 最終実行結果を  $\mathcal{A}$  へ返す.

**Challenge Phase:**  $\mathcal{A}$  は長さの等しい2つの平文  $m_0, m_1$ , ターゲット復号者アクセス構造  $A_d^*$  を  $CH$  へ送る.  $CH$  は,  $b \xleftarrow{U} \{0, 1\}$ ,  $CE.Enc(PK, m_b, A_d^*) \rightarrow C^*$  を実行し,  $C^*$  を  $\mathcal{A}$  に送る.

**Query Phase 2:**  $\mathcal{A}$  は以下のオラクルに対してクエリを任意回数発行することができる.

**秘密鍵生成:** Query Phase 1 と同様. 但し,  $S \in A_d^*$  を満たす  $S$  のクエリは禁止.

**復号:** Query Phase 1 と同様. 但し,  $C = C^*$  かつ  $S_r \in A_d^*$  を満たす  $(C, S_r)$  のクエリは禁止.

**Guess Phase:**  $\mathcal{A}$  はチャレンジビット  $b$  に対する推測として  $b' \in \{0, 1\}$  を出力.

CP-ABE 方式  $\Pi_{CE}$  に関する AP-IND-CCA 安全性ゲームにおいて攻撃者  $\mathcal{A}$  の優位性を次式で定義する.

$$\text{Adv}_{\Pi_{CE}, \mathcal{A}}^{\text{AP-IND-CCA}} = \left| \Pr[b' = b] - \frac{1}{2} \right| \quad (2.2)$$

**定義 2.2.** 全ての PPTA 攻撃者  $\mathcal{A}$  に対して, CP-ABE 方式  $\Pi_{CE}$  に関する AP-IND-CCA 安全性ゲームにおける  $\mathcal{A}$  の優位性  $\text{Adv}_{\Pi_{CE}, \mathcal{A}}^{\text{AP-IND-CCA}}$  が, セキュリティパラメータ  $k$  に関して無視できるほど小さい値であるならば, CP-ABE 方式  $\Pi_{CE}$  は AP-IND-CCA 安全である.

## 2.1.2 SP-IND-CCA 安全性

選択的述語モデルにおける適応的選択暗号文攻撃に対する暗号文識別不可能性 (Ciphertext-Indistinguishability under Adaptively Chosen Ciphertext Attack in the Selective Predicate Model, SP-IND-CCA) は, CP-ABE 方式が満たすべき秘匿性に関する安全性の中で, AP-IND-CCA(2.1.1 項) よりも弱いことが自明な安全性である. CP-ABE 方式  $\Pi_{CE}$  に関する SP-IND-CCA 安全性は, 攻撃者  $\mathcal{A}$  と挑戦者  $CH$  との間で行われる以下の SP-IND-CCA 安全性ゲームによって定義する.

**Init Phase:**  $CH$  はターゲット復号者アクセス構造  $A_d^*$  を  $CH$  へ送る.

**Setup Phase:**  $CH$  は  $CE.Setup(1^k, \mathcal{U}) \rightarrow (PK, MK)$  を実行し, システム公開鍵  $PK$  を  $\mathcal{A}$  に渡す.

**Query Phase 1:**  $\mathcal{A}$  は以下の各オラクルに対してクエリを任意回数発行することができる.

**秘密鍵生成:**  $\mathcal{A}$  は属性集合  $S_r \in (2^{\mathcal{U}} - \{\phi\})$  をクエリする。但し,  $S_r \in \mathbb{A}_d^*$  を満たす  $S_r$  のクエリは禁止とする。  $\mathcal{CH}$  は  $\text{CE.KeyGen}(\text{PK}, \text{MK}, S_r) \rightarrow \text{SK}_r$  を実行し  $\text{SK}_r$  を  $\mathcal{A}$  へ返す。

**復号:**  $\mathcal{A}$  は暗号文  $C$ , 属性集合  $S_r \in (2^{\mathcal{U}} - \{\phi\})$  をクエリする。  $\mathcal{CH}$  は  $\text{CE.KeyGen}(\text{PK}, \text{MK}, S_r) \rightarrow \text{SK}_r$ ,  $\text{CE.Dec}(\text{PK}, C, \text{SK}_r) \rightarrow m/\perp$  を実行し, 最終実行結果を  $\mathcal{A}$  へ返す。

**Challenge Phase:**  $\mathcal{A}$  は長さの等しい2つの平文  $m_0, m_1$  を  $\mathcal{CH}$  へ送る。  $\mathcal{CH}$  は,  $b \xleftarrow{\mathcal{U}} \{0, 1\}$ ,  $\text{CE.Enc}(\text{PK}, m_b, \mathbb{A}_d^*) \rightarrow C^*$  を実行し,  $C^*$  を  $\mathcal{A}$  に送る。

**Query Phase 2:**  $\mathcal{A}$  は以下のオラクルに対してクエリを任意回数発行することができる。

**秘密鍵生成:** Query Phase 1 と同様。但し,  $S \in \mathbb{A}_d^*$  を満たす  $S$  のクエリは禁止。

**復号:** Query Phase 1 と同様。但し,  $C = C^*$  かつ  $S_r \in \mathbb{A}_d^*$  を満たす  $(C, S_r)$  のクエリは禁止。

**Guess Phase:**  $\mathcal{A}$  はチャレンジビット  $b$  に対する推測として  $b' \in \{0, 1\}$  を出力。

CP-ABE 方式  $\Pi_{\text{CE}}$  に関する SP-IND-CCA 安全性ゲームにおいて攻撃者  $\mathcal{A}$  の優位性を次式で定義する。

$$\text{Adv}_{\Pi_{\text{CE}}, \mathcal{A}}^{\text{SP-IND-CCA}} = \left| \Pr[b' = b] - \frac{1}{2} \right| \quad (2.3)$$

**定義 2.3.** 全ての PPTA 攻撃者  $\mathcal{A}$  に対して, CP-ABE 方式  $\Pi_{\text{CE}}$  に関する SP-IND-CCA 安全性ゲームにおける  $\mathcal{A}$  の優位性  $\text{Adv}_{\Pi_{\text{CE}}, \mathcal{A}}^{\text{SP-IND-CCA}}$  が, セキュリティパラメータ  $k$  に関して無視できるほど小さい値であるならば, CP-ABE 方式  $\Pi_{\text{CE}}$  は SP-IND-CCA 安全である。

### 2.1.3 復号者アクセス構造開示性

CP-ABE 方式の“復号者アクセス構造開示性”は著者が独自に命名し, かつ定義した性質である。直感的には, 正当な暗号文から暗号文生成に使用された(あるいは, 暗号文と真に関連付けられた)復号者アクセス構造を完璧に復元できることを意味する。具体的には, 次の様に定義される。

**定義 2.4.** CP-ABE 方式が復号者アクセス構造開示性を満たすとは, PPTA  $\text{Disclose}_{\text{CE}}$  が存在し, 全ての  $k$ , 全ての  $\mathcal{U}$ , 全ての  $(\text{PK}, \text{MK}) \leftarrow \text{CE.Setup}(1^k, \mathcal{U})$ , 全ての  $m$ , 全ての  $\mathbb{A}_d$ , 全ての  $C \leftarrow \text{CE.Enc}(\text{PK}, m, \mathbb{A}_d)$  に対して, 以下の条件式が成立する場合を言う。

$$\Pr[\mathbb{A}'_d = \mathbb{A}_d | \mathbb{A}'_d \leftarrow \text{Disclose}_{\text{CE}}(\text{PK}, C)] = 1 \quad (2.4)$$

CP-ABE 方式の“復号者アクセス構造開示性”は特殊な性質ではなく極めて自然な性質である。事実として著者が知る限り、既存の CP-ABE 方式の大多数 [12][14][15][16][22][23][17][18][19][20][21] に関して、容易に定義 2.4 の条件を満たすような PPTA  $\text{Disclose}_{\text{CE}}$  を構成できる。

## 2.2 鍵ポリシー型属性ベース暗号

鍵ポリシー型属性ベース暗号 (Key-Policy Attribute-Based Encryption, 以降略称として“KP-ABE”と表記することもある) は、秘密鍵とポリシー (アクセス構造), 暗号文 (平文) と属性集合 (復号者属性集合) がそれぞれ対応する, 属性ベース暗号である。

KP-ABE は以下の 4 つのアルゴリズムから構成される。

$\text{KE.Setup}(1^k, \mathcal{U}) \rightarrow (\text{PK}, \text{MK})$

セットアップアルゴリズム  $\text{KE.Setup}$  は, セキュリティパラメータ  $k$ , 属性の全体集合  $\mathcal{U}$  を入力とし, システム公開鍵  $\text{PK}$  とマスター秘密鍵  $\text{MK}$  を出力する。

$\text{KE.KeyGen}(\text{PK}, \text{MK}, \mathbb{A}_r) \rightarrow \text{SK}_r$

鍵生成アルゴリズム  $\text{KE.KeyGen}$  は, 公開鍵  $\text{PK}$ , マスター秘密鍵  $\text{MK}$ , 受信者のポリシー  $\mathbb{A}_r$  を入力とし, 受信者用秘密鍵  $\text{SK}_r$  を出力する。

$\text{KE.Enc}(\text{PK}, m, S_d) \rightarrow C$

暗号化アルゴリズム  $\text{KE.Enc}$  は, システム公開鍵  $\text{PK}$ , 平文  $m$ , 復号者属性集合  $S_d$  を入力とし, 暗号文  $C$  を出力する。

$\text{KE.Dec}(\text{PK}, C, \text{SK}_r) \rightarrow m / \perp$

復号アルゴリズム  $\text{KE.Dec}$  は, システム公開鍵  $\text{PK}$ , 暗号文  $C$ , 受信者用秘密鍵  $\text{SK}_r$  を入力とし, 平文  $m \in \mathcal{M}$  または“復号不可”を表す特別な記号  $\perp$  を出力する。

**正当性 (Correctness)** KP-ABE 方式は正当であることが求められる。KP-ABE 方式が正当であるとは, 全ての  $k$ , 全ての  $\mathcal{U}$ , 全ての  $(\text{PK}, \text{MK}) \leftarrow \text{KE.Setup}(1^k, \mathcal{U})$ , 全ての  $\mathbb{A}_r$ , 全ての  $\text{SK}_r \leftarrow \text{KE.KeyGen}(\text{PK}, \text{MK}, \mathbb{A}_r)$ , 全ての  $S_d \in \mathbb{A}_r$  を満たす全ての  $S_d$ , 全ての  $C \leftarrow \text{KE.Enc}(\text{PK}, m, S_d)$  に対して, 次の等式が成立する場合を言う。

$$\Pr[\text{KE.Dec}(\text{PK}, C, \text{SK}_r) = m] = 1 \quad (2.5)$$

### 2.2.1 AA-IND-CCA 安全性

適応的属性モデルにおける適応的選択暗号文攻撃に対する暗号文識別不可能性 (Ciphertext-Indistinguishability under Adaptively Chosen Ciphertext Attack in the Adaptive Attribute Model, AA-IND-CCA) は, KP-ABE 方式が満たすべき秘匿性に関する安全性の中で, 最も強い安全性と考えられている。KP-ABE 方式  $\Pi_{\text{KE}}$  に関する AA-IND-CCA 安全性は, 攻撃者  $\mathcal{A}$  と挑戦者  $\mathcal{CH}$  との間で行われる以下の AA-IND-CCA 安全性ゲームによって定義する。

**Setup Phase:**  $\mathcal{CH}$  は  $\text{KE.Setup}(1^k, \mathcal{U}) \rightarrow (\text{PK}, \text{MK})$  を実行し、システム公開鍵  $\text{PK}$  を  $\mathcal{A}$  に渡す。

**Query Phase 1:**  $\mathcal{A}$  は以下の各オラクルに対してクエリを任意回数発行することができる。

**秘密鍵生成:**  $\mathcal{A}$  は  $\mathbb{A}_r$  をクエリする。但し、Challenge Phase で選択するターゲット復号者属性集合  $S_d^*$  に対して、 $S_d^* \in \mathbb{A}_r$  を満たす  $\mathbb{A}_r$  のクエリは禁止とする。 $\mathcal{CH}$  は  $\text{KE.KeyGen}(\text{PK}, \text{MK}, \mathbb{A}_r) \rightarrow \text{SK}_r$  を実行し  $\text{SK}_r$  を  $\mathcal{A}$  へ返す。

**復号:**  $\mathcal{A}$  は暗号文  $C$ 、ポリシー  $\mathbb{A}_r$  をクエリする。 $\mathcal{CH}$  は  $\text{KE.KeyGen}(\text{PK}, \text{MK}, \mathbb{A}_r) \rightarrow \text{SK}_r$ 、 $\text{KE.Dec}(\text{PK}, C, \text{SK}_r) \rightarrow m/\perp$  を実行し、最終実行結果を  $\mathcal{A}$  へ返す。

**Challenge Phase:**  $\mathcal{A}$  は長さの等しい2つの平文  $m_0, m_1$ 、ターゲット復号者属性集合  $S_d^*$  を  $\mathcal{CH}$  へ送る。 $\mathcal{CH}$  は、 $b \xleftarrow{\text{U}} \{0, 1\}$ 、 $\text{KE.Enc}(\text{PK}, m_b, S_d^*) \rightarrow C^*$  を実行し、 $C^*$  を  $\mathcal{A}$  に送る。

**Query Phase 2:**  $\mathcal{A}$  は以下のオラクルに対してクエリを任意回数発行することができる。

**秘密鍵生成:** Query Phase 1 と同様。但し、 $S_d^* \in \mathbb{A}_r$  を満たす  $\mathbb{A}_r$  のクエリは禁止。

**復号:** Query Phase 1 と同様。但し、 $C = C^*$  かつ  $S_d^* \in \mathbb{A}_r$  を満たす  $(C, \mathbb{A}_r)$  のクエリは禁止。

**Guess Phase:**  $\mathcal{A}$  はチャレンジビット  $b$  に対する推測として  $b' \in \{0, 1\}$  を出力。

KP-ABE 方式  $\Pi_{\text{KE}}$  に関する AA-IND-CCA 安全性ゲームにおいて攻撃者  $\mathcal{A}$  の優位性を次式で定義する。

$$\text{Adv}_{\Pi_{\text{KE}}, \mathcal{A}}^{\text{AA-IND-CCA}} = |\Pr[b' = b] - \frac{1}{2}| \quad (2.6)$$

**定義 2.5.** 全ての PPTA 攻撃者  $\mathcal{A}$  に対して、KP-ABE 方式  $\Pi_{\text{KE}}$  に関する AA-IND-CCA 安全性ゲームにおける  $\mathcal{A}$  の優位性  $\text{Adv}_{\Pi_{\text{KE}}, \mathcal{A}}^{\text{AA-IND-CCA}}$  が、セキュリティパラメータ  $k$  に関して無視できるほど小さい値であるならば、KP-ABE 方式  $\Pi_{\text{KE}}$  は AA-IND-CCA 安全である。

## 2.2.2 SA-IND-CCA 安全性

選択的属性モデルにおける適応的選択暗号文攻撃に対する暗号文識別不可能性 (Ciphertext-Indistinguishability under Adaptively Chosen Ciphertext Attack in the Selective Attribute Model, SA-IND-CCA) は, KP-ABE 方式が満たすべき秘匿性に関する安全性の中で, AA-IND-CCA(2.2.1 項) よりも弱いことが自明な安全性である. KP-ABE 方式  $\Pi_{\text{KE}}$  に関する SA-IND-CCA 安全性は, 攻撃者  $\mathcal{A}$  と挑戦者  $\mathcal{CH}$  との間で行われる以下の SA-IND-CCA 安全性ゲームによって定義する.

**Init Phase:**  $\mathcal{CH}$  はターゲット復号者属性集合  $S_d^*$  を  $\mathcal{CH}$  へ送る.

**Setup Phase:**  $\mathcal{CH}$  は  $\text{KE.Setup}(1^k, \mathcal{U}) \rightarrow (\text{PK}, \text{MK})$  を実行し, システム公開鍵  $\text{PK}$  を  $\mathcal{A}$  に渡す.

**Query Phase 1:**  $\mathcal{A}$  は以下の各オラクルに対してクエリを任意回数発行することができる.

**秘密鍵生成:**  $\mathcal{A}$  はポリシー  $A_r$  をクエリする. 但し, Challenge Phase で選択するターゲット復号者属性集合  $S_d^*$  に対して,  $S_d^* \in A_r$  を満たす  $A_r$  のクエリは禁止とする.  $\mathcal{CH}$  は  $\text{KE.KeyGen}(\text{PK}, \text{MK}, A_r) \rightarrow \text{SK}_r$  を実行し  $\text{SK}_r$  を  $\mathcal{A}$  へ返す.

**復号:**  $\mathcal{A}$  は暗号文  $C$ , ポリシー  $A_r$  をクエリする.  $\mathcal{CH}$  は  $\text{KE.KeyGen}(\text{PK}, \text{MK}, A_r) \rightarrow \text{SK}_r$ ,  $\text{KE.Dec}(\text{PK}, C, \text{SK}_r) \rightarrow m/\perp$  を実行し, 最終実行結果を  $\mathcal{A}$  へ返す.

**Challenge Phase:**  $\mathcal{A}$  は長さの等しい2つの平文  $m_0, m_1$  を  $\mathcal{CH}$  へ送る.  $\mathcal{CH}$  は,  $b \leftarrow \bigcup \{0, 1\}$ ,  $\text{KE.Enc}(\text{PK}, m_b, S_d^*) \rightarrow C^*$  を実行し,  $C^*$  を  $\mathcal{A}$  に送る.

**Query Phase 2:**  $\mathcal{A}$  は以下のオラクルに対してクエリを任意回数発行することができる.

**秘密鍵生成:** Query Phase 1 と同様. 但し,  $S_d^* \in A_r$  を満たす  $A_r$  のクエリは禁止.

**復号:** Query Phase 1 と同様. 但し,  $C = C^*$  かつ  $S_d^* \in A_r$  を満たす  $(C, A_r)$  のクエリは禁止.

**Guess Phase:**  $\mathcal{A}$  はチャレンジビット  $b$  に対する推測として  $b' \in \{0, 1\}$  を出力.

KP-ABE 方式  $\Pi_{\text{KE}}$  に関する SA-IND-CCA 安全性ゲームにおいて攻撃者  $\mathcal{A}$  の優位性を次式で定義する.

$$\text{Adv}_{\Pi_{\text{KE}}, \mathcal{A}}^{\text{SA-IND-CCA}} = |\Pr[b' = b] - \frac{1}{2}| \quad (2.7)$$

**定義 2.6.** 全ての  $PPTA$  攻撃者  $\mathcal{A}$  に対して,  $KP\text{-}ABE$  方式  $\Pi_{KE}$  に関する  $SA\text{-}IND\text{-}CCA$  安全性ゲームにおける  $\mathcal{A}$  の優位性  $Adv_{\Pi_{KE}, \mathcal{A}}^{SA\text{-}IND\text{-}CCA}$  が, セキュリティパラメータ  $k$  に関して無視できるほど小さい値であるならば,  $KP\text{-}ABE$  方式  $\Pi_{KE}$  は  $SA\text{-}IND\text{-}CCA$  安全である.

### 2.2.3 復号者属性集合開示性

$KP\text{-}ABE$  方式の“復号者属性集合開示性”は著者が独自に命名し, かつ定義した性質である. 直感的には, 正当な暗号文から暗号文生成に使用された (あるいは, 暗号文と真に関連付けられた) 復号者属性集合を完璧に復元できることを意味する. 具体的には, 次の様に定義される.

**定義 2.7.**  $KP\text{-}ABE$  方式が復号者属性集合開示性を満たすとは,  $PPTA\text{ Disclose}_{KE}$  が存在し, 全ての  $k$ , 全ての  $\mathcal{U}$ , 全ての  $(PK, MK) \leftarrow KE.Setup(1^k, \mathcal{U})$ , 全ての  $m$ , 全ての  $S_d \in (2^{\mathcal{U}} - \{\phi\})$ , 全ての  $C \leftarrow KE.Enc(PK, m, S_d)$  に対して, 以下の条件式が成立する場合を言う.

$$\Pr[S'_d = S_d | S'_d \leftarrow Disclose_{KE}(PK, C)] = 1 \quad (2.8)$$

$KP\text{-}ABE$  方式の“復号者属性集合開示性”は特殊な性質ではなく極めて自然な性質である. 事実として著者が知る限り, 既存の  $KP\text{-}ABE$  方式の多数 [22][23] に関して, 容易に定義 2.7 の条件を満たすような  $PPTA\text{ Disclose}_{KE}$  を構成できる.

## 2.3 暗号文ポリシー型属性ベース鍵カプセル化メカニズム

鍵カプセル化メカニズム (Key Encapsulation Mechanism, 以降略称として“ $KEM$ ”と表記することもある) は, (共通鍵暗号と比べた場合に) 効率の悪さが問題になる公開鍵暗号といわゆる“鍵配送問題”が生じる共通鍵暗号の両暗号の欠点を補う目的で考案されたハイブリッド暗号において, セッション鍵を共有するための公開鍵暗号方式による処理部分を定式化したものである.

そして, 暗号文ポリシー型属性ベース鍵カプセル化アルゴリズム (Ciphertext-Policy Attribute-Based Key Encapsulation Mechanism, 以降略称として“ $CP\text{-}ABKEM$ ”と表記することもある) は,  $KEM$  の属性ベース版であり, 鍵暗号文にポリシーと呼ばれる復号者条件を関連付けて, ポリシーを満たす属性集合を持つ人物だけが, 鍵暗号文を正しく復号してセッション鍵を入手することができる暗号方式である.  $CP\text{-}ABKEM$  は, 鍵暗号文とポリシー (復号者アクセス構造), 秘密鍵と属性集合がそれぞれ対応する.

$CP\text{-}ABKEM$  方式は以下の 4 つのアルゴリズムから構成される.

$CK.Setup(1^k, \mathcal{U}) \rightarrow (PK, MK)$

セットアップアルゴリズム  $CK.Setup$  は, セキュリティパラメータ  $k$ , 属性の全体集合  $\mathcal{U}$  を入力とし, システム公開鍵  $PK$  とマスター秘密鍵  $MK$  を出力する.

$CK.KeyGen(PK, MK, S_r) \rightarrow SK_r$

鍵生成アルゴリズム  $CK.KeyGen$  は, システム公開鍵  $PK$ , マスター秘密鍵  $MK$ , 受信者用属性集合  $S_r \in (2^{\mathcal{U}} - \{\phi\})$  を入力とし, 受信者用秘密鍵  $SK_r$  を出力する.

**CK.Encap**(PK,  $\mathbb{A}_d$ )  $\rightarrow$  ( $K, C_K$ )

鍵カプセル化アルゴリズム **CK.Encap** は, システム公開鍵 PK, 復号者アクセス構造  $\mathbb{A}_d$  を入力とし, 鍵  $K \in \mathcal{K}$  とその暗号文  $C_K$  を出力する.

**CK.Decap**(PK,  $C_K, SK_r$ )  $\rightarrow K / \perp$

鍵復号アルゴリズム **CK.Decap** は, システム公開鍵 PK, 鍵暗号文  $C_K$ , 受信者用秘密鍵  $SK_r$  を入力とし, 鍵  $K \in \mathcal{K}$  または復号不可を表す特別な記号  $\perp$  を出力する.

**正当性 (Correctness)** CP-ABKEM 方式は正当であることが求められる. CP-ABKEM 方式が正当であるとは, 全ての  $k$ , 全ての  $\mathcal{U}$ , 全ての  $(PK, MK) \leftarrow \text{CK.Setup}(1^k, \mathcal{U})$ , 全ての  $S_r \in (2^{\mathcal{U}} - \{\phi\})$ , 全ての  $SK_r \leftarrow \text{CK.KeyGen}(PK, MK, S_r)$ ,  $S_r \in \mathbb{A}_d$  を満たす全ての  $\mathbb{A}_d$ , 全ての  $(K, C_K) \leftarrow \text{CK.Encap}(PK, \mathbb{A}_d)$  に対して, 次の等式が成立する場合を言う.

$$\Pr[\text{CK.Decap}(PK, C_K, SK_r) = K] = 1 \quad (2.9)$$

### 2.3.1 AP-IND-CCA 安全性

適応的述語モデルにおける適応的選択暗号文攻撃に対する暗号文識別不可能性 (Ciphertext-Indistinguishability under Adaptively Chosen Ciphertext Attack in the Adaptive Predicate Model, AP-IND-CCA) は, CP-ABKEM 方式が満たすべき秘匿性に関する安全性の中で, 最も強い安全性と考えられている. CP-ABKEM 方式  $\Pi_{\text{CK}}$  に関する AP-IND-CCA 安全性は, 攻撃者  $\mathcal{A}$  と挑戦者  $\mathcal{CH}$  との間で行われる以下の AP-IND-CCA 安全性ゲームによって定義する.

**Setup Phase:**  $\mathcal{CH}$  は  $\text{CK.Setup}(1^k, \mathcal{U}) \rightarrow (PK, MK)$  を実行し, システム公開鍵 PK を  $\mathcal{A}$  に送る.

**Query Phase 1:**  $\mathcal{A}$  は以下の各オラクルに対してクエリを任意回数発行できる.

**秘密鍵生成:**  $\mathcal{A}$  は属性集合  $S_r \in (2^{\mathcal{U}} - \{\phi\})$  をクエリする. 但し, Challenge フェーズで選択するターゲット復号者アクセス構造  $\mathbb{A}_d^*$  に関して,  $S_r \in \mathbb{A}_d^*$  を満たす  $S_r$  のクエリは禁止.  $\mathcal{CH}$  は  $\text{CK.KeyGen}(PK, MK, S_r) \rightarrow SK_r$  を実行し  $SK_r$  を  $\mathcal{A}$  へ送る.

**鍵復号:**  $\mathcal{A}$  は鍵暗号文  $C_K$ , 属性集合  $S_r \in (2^{\mathcal{U}} - \{\phi\})$  をクエリする.  $\mathcal{CH}$  は  $\text{CK.KeyGen}(PK, MK, S_r) \rightarrow SK_r$ ,  $\text{CK.Decap}(PK, C_K, SK_r) \rightarrow K / \perp$  を実行し最終実行結果を  $\mathcal{A}$  へ送る.

**Challenge Phase:**  $\mathcal{A}$  はターゲット復号者アクセス構造  $\mathbb{A}_d^*$  を  $\mathcal{CH}$  へ送る.  $\mathcal{CH}$  は  $\text{CK.Encap}(PK, \mathbb{A}_d^*) \rightarrow (K_1, C_K^*), K_0 \stackrel{\mathcal{U}}{\leftarrow} \mathcal{K}, b \stackrel{\mathcal{U}}{\leftarrow} \{0, 1\}$  を実行し,  $(K_b, C_K^*)$

を  $\mathcal{A}$  に渡す.

**Query Phase 2:**  $\mathcal{A}$  は以下の各オラクルに対してクエリを任意回数発行できる.

**秘密鍵生成:** Query Phase 1 と同様. 但し,  $S_r \in \mathbb{A}_d^*$  を満たす  $S_r$  のクエリは禁止.

**鍵復号:** Query Phase 1 と同様. 但し,  $C_K = C_K^*$  かつ  $S_r \in \mathbb{A}_d^*$  を満たす  $(C_K, S_r)$  のクエリは禁止.

**Guess Phase:**  $\mathcal{A}$  はチャレンジビット  $b$  に対する推測ビットとして  $b' \in \{0, 1\}$  を出力.

CP-ABKEM 方式  $\Pi_{\text{CK}}$  に関する AP-IND-CCA 安全性ゲームにおいて攻撃者  $\mathcal{A}$  の優位性を次式で定義する.

$$\text{Adv}_{\Pi_{\text{CK}}, \mathcal{A}}^{\text{AP-IND-CCA}} = |\Pr[b' = b] - \frac{1}{2}| \quad (2.10)$$

**定義 2.8.** 全ての PPTA 攻撃者  $\mathcal{A}$  に対して, CP-ABKEM 方式  $\Pi_{\text{CK}}$  に関する AP-IND-CCA 安全性ゲームにおける  $\mathcal{A}$  の優位性  $\text{Adv}_{\Pi_{\text{CK}}, \mathcal{A}}^{\text{AP-IND-CCA}}$  が, セキュリティパラメータ  $k$  に関して無視できるほど小さい値であるならば, CP-ABKEM 方式  $\Pi_{\text{CK}}$  は AP-IND-CCA 安全である.

### 2.3.2 SP-IND-CCA 安全性

選択的述語モデルにおける適応的選択暗号文攻撃に対する暗号文識別不可能性 (Ciphertext-Indistinguishability under Adaptively Chosen Ciphertext Attack in the Selective Predicate Model, SP-IND-CCA) は, CP-ABKEM 方式が満たすべき秘匿性に関する安全性の中で, AP-IND-CCA(2.3.1 項) よりも弱いことが自明な安全性である. CP-ABKEM 方式  $\Pi_{\text{CK}}$  に関する AP-IND-CCA 安全性は, 攻撃者  $\mathcal{A}$  と挑戦者  $\mathcal{CH}$  との間で行われる以下の SP-IND-CCA 安全性ゲームによって定義する.

**Init Phase:**  $\mathcal{CH}$  はターゲット復号者アクセス構造  $\mathbb{A}_d^*$  を  $\mathcal{CH}$  へ送る.

**Setup Phase:**  $\mathcal{CH}$  は  $\text{CK.Setup}(1^k, \mathcal{U}) \rightarrow (\text{PK}, \text{MK})$  を実行し, システム公開鍵  $\text{PK}$  を  $\mathcal{A}$  に送る.

**Query Phase 1:**  $\mathcal{A}$  は以下の各オラクルに対してクエリを任意回数発行できる.

**秘密鍵生成:**  $\mathcal{A}$  は属性集合  $S_r \in (2^{\mathcal{U}} - \{\emptyset\})$  をクエリする. 但し,  $S_r \in \mathbb{A}_d^*$  を満たす  $S_r$  のクエリは禁止.  $\mathcal{CH}$  は  $\text{CK.KeyGen}(\text{PK}, \text{MK}, S_r) \rightarrow \text{SK}_r$

を実行し  $SK_r$  を  $\mathcal{A}$  へ送る.

**鍵復号:**  $\mathcal{A}$  は鍵暗号文  $C_K$ , 属性集合  $S_r \in (2^{\mathcal{U}} - \{\emptyset\})$  をクエリする.  $\mathcal{CH}$  は  $CK.KeyGen(PK, MK, S_r) \rightarrow SK_r$ ,  $CK.Decap(PK, C_K, SK_r) \rightarrow K / \perp$  を実行し最終実行結果を  $\mathcal{A}$  へ送る.

**Challenge Phase:**  $\mathcal{CH}$  は  $CK.Encap(PK, A_d^*) \rightarrow (K_1, C_K^*), K_0 \xleftarrow{\mathcal{U}} \mathcal{K}, b \xleftarrow{\mathcal{U}} \{0, 1\}$  を実行し,  $(K_b, C_K^*)$  を  $\mathcal{A}$  に渡す.

**Query Phase 2:**  $\mathcal{A}$  は以下の各オラクルに対してクエリを任意回数発行できる.

**秘密鍵生成:** Query Phase 1 と同様. 但し,  $S_r \in A_d^*$  を満たす  $S_r$  のクエリは禁止.

**鍵復号:** Query Phase 1 と同様. 但し,  $C_K = C_K^*$  かつ  $S_r \in A_d^*$  を満たす  $(C_K, S_r)$  のクエリは禁止.

**Guess Phase:**  $\mathcal{A}$  はチャレンジビット  $b$  に対する推測ビットとして  $b' \in \{0, 1\}$  を出力.

CP-ABKEM 方式  $\Pi_{CK}$  に関する SP-IND-CCA 安全性ゲームにおいて攻撃者  $\mathcal{A}$  の優位性を次式で定義する.

$$\text{Adv}_{\Pi_{CK}, \mathcal{A}}^{\text{SP-IND-CCA}} = |\Pr[b' = b] - \frac{1}{2}| \quad (2.11)$$

**定義 2.9.** 全ての PPTA 攻撃者  $\mathcal{A}$  に対して, CP-ABKEM 方式  $\Pi_{CK}$  に関する AP-IND-CCA 安全性ゲームにおける  $\mathcal{A}$  の優位性  $\text{Adv}_{\Pi_{CK}, \mathcal{A}}^{\text{SP-IND-CCA}}$  が, セキュリティパラメータ  $k$  に関して無視できるほど小さい値であるならば, CP-ABKEM 方式  $\Pi_{CK}$  は SP-IND-CCA 安全である.

### 2.3.3 復号者アクセス構造開示性

CP-ABKEM 方式の“復号者アクセス構造開示性”は著者が独自に命名し, かつ定義した性質である. 直感的には, 正当な鍵暗号文から鍵暗号文生成に使用された (あるいは, 鍵暗号文と真に関連付けられた) 復号者アクセス構造を完璧に復元できることを意味する. 具体的には, 次の様に定義される.

**定義 2.10.** CP-ABKEM 方式が復号者アクセス構造開示性を満たすとは, PPTA  $\text{Disclose}_{CK}$  が存在し, 全ての  $k$ , 全ての  $\mathcal{U}$ , 全ての  $(PK, MK) \leftarrow CK.Setup(1^k, \mathcal{U})$ , 全ての  $A_d$ , 全ての  $(K, C_K) \leftarrow CK.Encap(PK, A_d)$  に対して, 以下の条件式が成立する場合を言う.

$$\Pr[A'_d = A_d | A'_d \leftarrow \text{Disclose}_{CK}(PK, C_K)] = 1 \quad (2.12)$$

CP-ABKEM 方式の“復号者アクセス構造開示性”は特殊な性質ではなく極めて自然な性質である。事実として著者が知る限り、既存の CP-ABKEM 方式のほぼ全て [34][35][36][37] に関して、容易に定義 2.10 の条件を満たすような PPTA  $\text{Disclose}_{\text{CK}}$  を構成できる。

## 2.4 鍵ポリシー型属性ベース鍵カプセル化メカニズム

鍵ポリシー型属性ベース鍵カプセル化メカニズム (Key-Policy Attribute-Based Key Encapsulation Mechanism, 以降略称として“KP-ABKEM”と表記することもある) は、秘密鍵とポリシー (アクセス構造), 鍵暗号文 (平文) と属性集合 (復号者属性集合) がそれぞれ対応する, 属性ベース鍵カプセル化メカニズムである。

KP-ABE は以下の 4 つのアルゴリズムから構成される。

$\text{KK.Setup}(1^k, \mathcal{U}) \rightarrow (\text{PK}, \text{MK})$

セットアップアルゴリズム  $\text{KK.Setup}$  は, セキュリティパラメータ  $k$ , 属性の全体集合  $\mathcal{U}$  を入力とし, システム公開鍵  $\text{PK}$  とマスター秘密鍵  $\text{MK}$  を出力する。

$\text{KK.KeyGen}(\text{PK}, \text{MK}, \mathbb{A}_r) \rightarrow \text{SK}_r$

鍵生成アルゴリズム  $\text{KK.KeyGen}$  は, システム公開鍵  $\text{PK}$ , マスター秘密鍵  $\text{MK}$ , 受信者のポリシー  $\mathbb{A}_r$  を入力とし, 受信者用秘密鍵  $\text{SK}_r$  を出力する。

$\text{KK.Encap}(\text{PK}, S_d) \rightarrow (K, C_K)$

鍵カプセル化アルゴリズム  $\text{KK.Encap}$  は, システム公開鍵  $\text{PK}$ , 復号者属性集合  $S_d$  を入力とし, 鍵  $K \in \mathcal{K}$  とその暗号文  $C_K$  を出力する。

$\text{KK.Decap}(\text{PK}, C_K, \text{SK}_r) \rightarrow K / \perp$

鍵復号アルゴリズム  $\text{KK.Decap}$  は, システム公開鍵  $\text{PK}$ , 鍵暗号文  $C_K$ , 受信者用秘密鍵  $\text{SK}_r$  を入力とし, 鍵  $K \in \mathcal{K}$  または復号不可を表す特別な記号  $\perp$  を出力する。

**正当性 (Correctness)** KP-ABKEM 方式は正当であることが求められる。KP-ABKEM 方式が正当であるとは, 全ての  $k$ , 全ての  $\mathcal{U}$ , 全ての  $(\text{PK}, \text{MK}) \leftarrow \text{KK.Setup}(1^k, \mathcal{U})$ , 全ての  $\mathbb{A}_r$ , 全ての  $\text{SK}_r \leftarrow \text{KK.KeyGen}(\text{PK}, \text{MK}, \mathbb{A}_r)$ ,  $S_d \in \mathbb{A}_r$  を満たす全ての  $S_d$ , 全ての  $(K, C_K) \leftarrow \text{KK.Encap}(\text{PK}, S_d)$  に対して, 次の等式が成立する場合を言う。

$$\Pr[\text{KK.Decap}(\text{PK}, C_K, \text{SK}_r) = K] = 1 \quad (2.13)$$

### 2.4.1 AA-IND-CCA 安全性

適応的属性モデルにおける適応的選択暗号文攻撃に対する暗号文識別不可能性 (Ciphertext-Indistinguishability under Adaptively Chosen Ciphertext Attack in the Adaptive Attribute Model, AA-IND-CCA) は, KP-ABKEM 方式が満たすべき秘匿性に関する

る安全性の中で、最も強い安全性と考えられている。KP-ABKEM 方式  $\Pi_{\text{KK}}$  に関する AA-IND-CCA 安全性は、攻撃者  $\mathcal{A}$  と挑戦者  $\mathcal{CH}$  との間で行われる以下の AA-IND-CCA 安全性ゲームによって定義する。

**Setup Phase:**  $\mathcal{CH}$  は  $\text{KK.Setup}(1^k, \mathcal{U}) \rightarrow (\text{PK}, \text{MK})$  を実行し、システム公開鍵  $\text{PK}$  を  $\mathcal{A}$  に送る。

**Query Phase 1:**  $\mathcal{A}$  は以下の各オラクルに対してクエリを任意回数発行できる。

**秘密鍵生成:**  $\mathcal{A}$  はポリシー  $\mathbb{A}_r$  をクエリする。但し、Challenge フェーズで選択するターゲット復号者属性集合  $S_d^*$  に関して、 $S_d^* \in \mathbb{A}_r$  を満たす  $\mathbb{A}_r$  のクエリは禁止。  $\mathcal{CH}$  は  $\text{KK.KeyGen}(\text{PK}, \text{MK}, \mathbb{A}_r) \rightarrow \text{SK}_r$  を実行し  $\text{SK}_r$  を  $\mathcal{A}$  へ送る。

**鍵復号:**  $\mathcal{A}$  は鍵暗号文  $C_K$ 、ポリシー  $\mathbb{A}_r$  をクエリする。  $\mathcal{CH}$  は  $\text{KK.KeyGen}(\text{PK}, \text{MK}, \mathbb{A}_r) \rightarrow \text{SK}_r$ ,  $\text{KK.Decap}(\text{PK}, C_K, \text{SK}_r) \rightarrow K / \perp$  を実行し最終実行結果を  $\mathcal{A}$  へ送る。

**Challenge Phase:**  $\mathcal{A}$  はターゲット復号者属性集合  $S_d^*$  を  $\mathcal{CH}$  へ送る。  $\mathcal{CH}$  は  $\text{KK.Encap}(\text{PK}, S_d^*) \rightarrow (K_1, C_K^*), K_0 \xleftarrow{\mathcal{U}} \mathcal{K}, b \xleftarrow{\mathcal{U}} \{0, 1\}$  を実行し、 $(K_b, C_K^*)$  を  $\mathcal{A}$  に渡す。

**Query Phase 2:**  $\mathcal{A}$  は以下の各オラクルに対してクエリを任意回数発行できる。

**秘密鍵生成:** Query Phase 1 と同様。但し、 $S_d^* \in \mathbb{A}_r$  を満たす  $\mathbb{A}_r$  のクエリは禁止。

**鍵復号:** Query Phase 1 と同様。但し、 $C_K = C_K^*$  かつ  $S_d^* \in \mathbb{A}_r$  を満たす  $(C_K, \mathbb{A}_r)$  のクエリは禁止。

**Guess Phase:**  $\mathcal{A}$  はチャレンジビット  $b$  に対する推測ビットとして  $b' \in \{0, 1\}$  を出力。

KP-ABKEM 方式  $\Pi_{\text{KK}}$  に関する AA-IND-CCA 安全性ゲームにおいて攻撃者  $\mathcal{A}$  の優位性を次式で定義する。

$$\text{Adv}_{\Pi_{\text{KK}}, \mathcal{A}}^{\text{AA-IND-CCA}} = \left| \Pr[b' = b] - \frac{1}{2} \right| \quad (2.14)$$

**定義 2.11.** 全ての PPTA 攻撃者  $\mathcal{A}$  に対して、KP-ABKEM 方式  $\Pi_{\text{KK}}$  に関する AA-IND-CCA 安全性ゲームにおける  $\mathcal{A}$  の優位性  $\text{Adv}_{\Pi_{\text{KK}}, \mathcal{A}}^{\text{AA-IND-CCA}}$  が、セキュリティパラメータ  $k$  に関して無視できるほど小さい値であるならば、KP-ABKEM 方式  $\Pi_{\text{KK}}$  は AA-IND-CCA 安全である。

## 2.4.2 SA-IND-CCA 安全性

選択的属性モデルにおける適応的選択暗号文攻撃に対する暗号文識別不可能性 (Ciphertext-Indistinguishability under Adaptively Chosen Ciphertext Attack in the Selective Attribute Model, SA-IND-CCA) は, KP-ABKEM 方式が満たすべき秘匿性に関する安全性の中で, AA-IND-CCA(2.4.1 項) よりも弱いことが自明な安全性である. KP-ABKEM 方式  $\Pi_{\text{KK}}$  に関する SA-IND-CCA 安全性は, 攻撃者  $\mathcal{A}$  と挑戦者  $\mathcal{CH}$  との間で行われる以下の SA-IND-CCA 安全性ゲームによって定義する.

**Init Phase:**  $\mathcal{CH}$  はターゲット復号者属性集合  $S_d^*$  を  $\mathcal{CH}$  へ送る.

**Setup Phase:**  $\mathcal{CH}$  は  $\text{KK.Setup}(1^k, \mathcal{U}) \rightarrow (\text{PK}, \text{MK})$  を実行し, システム公開鍵  $\text{PK}$  を  $\mathcal{A}$  に送る.

**Query Phase 1:**  $\mathcal{A}$  は以下の各オラクルに対してクエリを任意回数発行できる.

**秘密鍵生成:**  $\mathcal{A}$  はポリシー  $A_r$  をクエリする. 但し, Challenge フェーズで選択するターゲット復号者属性集合  $S_d^*$  に関して,  $S_d^* \in A_r$  を満たす  $A_r$  のクエリは禁止.  $\mathcal{CH}$  は  $\text{KK.KeyGen}(\text{PK}, \text{MK}, A_r) \rightarrow \text{SK}_r$  を実行し  $\text{SK}_r$  を  $\mathcal{A}$  へ送る.

**鍵復号:**  $\mathcal{A}$  は鍵暗号文  $C_K$ , ポリシー  $A_r$  をクエリする.  $\mathcal{CH}$  は  $\text{KK.KeyGen}(\text{PK}, \text{MK}, A_r) \rightarrow \text{SK}_r$ ,  $\text{KK.Decap}(\text{PK}, C_K, \text{SK}_r) \rightarrow K / \perp$  を実行し最終実行結果を  $\mathcal{A}$  へ送る.

**Challenge Phase:**  $\mathcal{CH}$  は  $\text{KK.Encap}(\text{PK}, S_d^*) \rightarrow (K_1, C_K^*), K_0 \xleftarrow{\mathcal{U}} \mathcal{K}, b \xleftarrow{\mathcal{U}} \{0, 1\}$  を実行し,  $(K_b, C_K^*)$  を  $\mathcal{A}$  に渡す.

**Query Phase 2:**  $\mathcal{A}$  は以下の各オラクルに対してクエリを任意回数発行できる.

**秘密鍵生成:** Query Phase 1 と同様. 但し,  $S_d^* \in A_r$  を満たす  $A_r$  のクエリは禁止.

**鍵復号:** Query Phase 1 と同様. 但し,  $C_K = C_K^*$  かつ  $S_d^* \in A_r$  を満たす  $(C_K, A_r)$  のクエリは禁止.

**Guess Phase:**  $\mathcal{A}$  はチャレンジビット  $b$  に対する推測ビットとして  $b' \in \{0, 1\}$  を出力.

KP-ABKEM 方式  $\Pi_{\text{KK}}$  に関する SA-IND-CCA 安全性ゲームにおいて攻撃者  $\mathcal{A}$  の優位性を次式で定義する.

$$\text{Adv}_{\Pi_{\text{KK}}, \mathcal{A}}^{\text{SA-IND-CCA}} = |\Pr[b' = b] - \frac{1}{2}| \quad (2.15)$$

**定義 2.12.** 全ての  $PPTA$  攻撃者  $\mathcal{A}$  に対して,  $KP-ABKEM$  方式  $\Pi_{KK}$  に関する  $SA-IND-CCA$  安全性ゲームにおける  $\mathcal{A}$  の優位性  $Adv_{\Pi_{KK}, \mathcal{A}}^{SA-IND-CCA}$  が, セキュリティパラメータ  $k$  に関して無視できるほど小さい値であるならば,  $KP-ABKEM$  方式  $\Pi_{KK}$  は  $SA-IND-CCA$  安全である.

### 2.4.3 復号者属性集合開示性

$KP-ABKEM$  方式の“復号者属性集合開示性”は著者が独自に命名し, かつ定義した性質である. 直感的には, 正当な鍵暗号文から鍵暗号文生成に使用された (あるいは, 鍵暗号文と真に関連付けられた) 復号者属性集合を完璧に復元できることを意味する. 具体的には, 次の様に定義される.

**定義 2.13.**  $KP-ABKEM$  方式が復号者属性集合開示性を満たすとは,  $PPTA$   $Disclose_{KK}$  が存在し, 全ての  $k$ , 全ての  $\mathcal{U}$ , 全ての  $(PK, MK) \leftarrow KK.Setup(1^k, \mathcal{U})$ , 全ての  $S_d \in (2^{\mathcal{U}} - \{\phi\})$ , 全ての  $(K, C_K) \leftarrow KK.Encap(PK, S_d)$  に対して, 以下の条件式が成立する場合を言う.

$$\Pr[S'_d = S_d | S'_d \leftarrow Disclose_{KK}(PK, C_K)] = 1 \quad (2.16)$$

$KP-ABKEM$  方式の“復号者属性集合開示性”は,  $KP-ABE$  方式の復号者属性集合開示性や,  $CP-ABKEM$  方式の復号者アクセス構造開示性が特殊な性質ではなく極めて自然な性質であったことから, それらと同様に極めて自然な性質であると考えられる.

## 2.5 データカプセル化メカニズム

データカプセル化メカニズム (Data Encapsulation Mechanism, 以降略称として“DEM”と表記することもある) は, 公開鍵暗号と共通鍵暗号の両暗号の欠点を解消する目的で考案されたハイブリッド暗号において, セッション鍵を使ってデータ (あるいは平文) の暗号化と暗号文の復号を行う共通鍵暗号方式による処理部分を定式化したものである.

DEM 方式は以下の 2 つのアルゴリズムから構成される.

$D.Encap(K, m) \rightarrow C$

データカプセル化アルゴリズム  $Encap$  は, 鍵  $K \in \mathcal{K}$  と平文  $m \in \mathcal{M}$  を入力とし, 暗号文  $C \in \mathcal{C}$  を出力する.

$D.Decap(K, C) \rightarrow m / \perp$

データ復号アルゴリズム  $Decap$  は, 鍵  $K \in \mathcal{K}$  と暗号文  $C \in \mathcal{C}$  を入力とし, 平文  $m \in \mathcal{M}$  または  $\perp$  を出力する.

**正当性 (Correctness)** DEM 方式は正当であることが求められる. DEM 方式が正当であるとは, 全ての  $K \in \mathcal{K}$ , 全ての  $m \in \mathcal{M}$ , 全ての  $C \leftarrow D.Encap(K, m)$  に対して, 次の等式が成立する場合を言う.

$$\Pr[D.Decap(K, C) = m] = 1 \quad (2.17)$$

### 2.5.1 IND-CCA 安全性

適応的選択暗号文攻撃に対する暗号文識別不可能性 (Ciphertext-Indistinguishability under Adaptively Chosen Ciphertext Attacks, IND-CCA) は, DEM 方式が満たすべき秘匿性に関する安全性である. DEM 方式  $\Pi_D$  に関する IND-CCA 安全性は, 攻撃者  $\mathcal{A}$  と挑戦者  $\mathcal{CH}$  との間で行われる以下の IND-CCA 安全性ゲームによって定義する.

**Setup Phase:**  $\mathcal{CH}$  は  $K \xleftarrow{\mathcal{U}} \mathcal{K}$  を実行する.

**Query Phase 1:**  $\mathcal{A}$  は以下の各オラクルに対してクエリを任意回数発行できる.

データ復号:  $\mathcal{A}$  は暗号文  $C \in \mathcal{C}$  をクエリする.  $\mathcal{CH}$  は  $D.\text{Decap}(K, C) \rightarrow m / \perp$  を実行し最終結果を  $\mathcal{A}$  へ送る.

**Challenge Phase:**  $\mathcal{A}$  は長さが等しい平文  $m_0 \in \mathcal{M}, m_1 \in \mathcal{M}$  を  $\mathcal{CH}$  へ送る.  $\mathcal{CH}$  は  $b \leftarrow \{0, 1\}, D.\text{Encap}(K, m_b) \rightarrow C^*$  を実行し,  $C^*$  を  $\mathcal{A}$  へ送る.

**Query Phase 2:**  $\mathcal{A}$  は以下の各オラクルに対してクエリを任意回数発行することができる.

データ復号: Query Phase 1 と同様. 但し,  $C^*$  のクエリは禁止.

**Guess Phase:**  $\mathcal{A}$  はチャレンジビット  $b$  に対する推測として  $b' \in \{0, 1\}$  を出力.

DEM 方式  $\Pi_D$  に関する IND-CCA 安全性ゲームにおいて攻撃者  $\mathcal{A}$  の優位性を次式で定義する.

$$\text{Adv}_{\Pi_D, \mathcal{A}}^{\text{IND-CCA}} = |\Pr[b' = b] - \frac{1}{2}| \quad (2.18)$$

**定義 2.14.** 全ての PPTA 攻撃者  $\mathcal{A}$  に対して, DEM 方式  $\Pi_D$  に関する IND-CCA 安全性ゲームにおける  $\mathcal{A}$  の優位性  $\text{Adv}_{\Pi_D, \mathcal{A}}^{\text{IND-CCA}}$  が, セキュリティパラメータ  $k$  に関して無視できるほど小さい値であるならば, DEM 方式  $\Pi_D$  は IND-CCA 安全である.

### 2.5.2 1対1対応性

DEM が 1対1対応であるとは, いかなる鍵  $K \in \mathcal{K}$ , いかなる平文  $m \in \mathcal{M}$  に関しても, 以下の等式を満たす暗号文  $C \in \mathcal{C}$  が高々一つしか存在しない場合を言う.

$$D.\text{Decap}(K, C) = m \quad (2.19)$$

[31]によると, 1対1対応である DEM 方式は, IND-CCA 安全なものを含め多数存在する.

## 2.6 署名ポリシー型属性ベース署名

署名ポリシー型属性ベース署名 (Signature-Policy Attribute-Based Signature, 以降略称として“SP-ABS”と表記することもある) では, 署名者はある平文に関して自身の属性集合が満たす様なポリシー (署名者アクセス構造) を指定した上で自身の秘密鍵を使って署名を作成し, 署名検証者は署名を検証する事でそのポリシー (署名者アクセス構造) を満たす属性集合を持つ人物が確実にその平文を作成した事実を確認できる. SP-ABS は, 署名とポリシー (署名者アクセス構造), 秘密鍵と属性集合がそれぞれ対応する, 属性ベース署名である.

SP-ABS 方式は以下の4つのアルゴリズムから構成される.

**SS.Setup**( $1^k, \mathcal{U}$ )  $\rightarrow$  (PK, MK)

セットアップアルゴリズム **SS.Setup** は, セキュリティパラメータ  $k$  と, 属性の全体集合  $\mathcal{U}$  を入力とし, システム公開鍵 PK とマスター秘密鍵 MK を出力する.

**SS.KeyGen**(PK, MK,  $S_s$ )  $\rightarrow$  SK $_s$

鍵生成アルゴリズム **SS.KeyGen** は, システム公開鍵 PK, マスター秘密鍵 MK, 送信者用属性集合  $S_s \in (2^{\mathcal{U}} - \{\emptyset\})$  を入力とし, 送信者用秘密鍵 SK $_s$  を出力する.

**SS.Sig**(PK,  $m$ , SK $_s$ ,  $\mathbb{A}_s$ )  $\rightarrow$   $\sigma$

署名生成アルゴリズム **SS.Sig** は, システム公開鍵 PK, 平文  $m \in \mathcal{M}$ , 秘密鍵 SK, 署名者アクセス構造  $\mathbb{A}_s$  を入力とし, 署名  $\sigma$  を出力する.

**SS.Ver**(PK,  $\sigma$ ,  $m$ ,  $\mathbb{A}_s$ )  $\rightarrow$  1 / 0

署名検証アルゴリズム **SS.Ver** は, システム公開鍵 PK, 署名  $\sigma$ , 平文  $m \in \mathcal{M}$ , 署名者アクセス構造  $\mathbb{A}_s$  を入力とし, 署名検証合格を表す“1”, あるいは不合格を表す“0”を出力する.

**正当性 (Correctness)** SP-ABS 方式は正当であることが求められる. SP-ABS 方式が正当であるとは, 全ての  $k$ , 全ての  $\mathcal{U}$ , 全ての  $(\text{PK}, \text{MK}) \leftarrow \text{SS.Setup}(1^k, \mathcal{U})$ , 全ての  $m \in \mathcal{M}$ , 全ての  $S_s \in (2^{\mathcal{U}} - \{\emptyset\})$ , 全ての  $\forall \text{SK}_s \leftarrow \text{SS.KeyGen}(\text{PK}, \text{MK}, S_s)$ ,  $S_s \in \mathbb{A}_s$  を満たす全ての  $\mathbb{A}_s$ , 全ての  $\sigma \leftarrow \text{SS.Sig}(\text{PK}, m, \text{SK}, \mathbb{A}_s)$  に対して, 次の等式が成立する場合を言う.

$$\Pr[\text{SS.Ver}(\text{PK}, \sigma, m, \mathbb{A}_s) = 1] = 1 \quad (2.20)$$

### 2.6.1 AP-sEUF-CMA 安全性

適応的述語モデルにおける適応的選択文書攻撃に対する署名強偽造不可能性 (Strongly Existentially Unforgeability of Signature under Adaptively Chosen Message Attacks in the Adaptive Predicate model, AP-sEUF-CMA) は, SP-ABS 方式が満たすべき完全性に関する安全性の中で, 最も強い安全性である. SP-ABS 方式  $\Pi_{\text{SS}}$  に関する AP-sEUF-CMA 安全性は, 攻撃者  $\mathcal{A}$  と挑戦者  $\mathcal{CH}$  との間で行われる以下の AP-sEUF-CMA 安全性ゲームによって定義する.

**Setup Phase:**  $\mathcal{CH}$  が  $\text{SS.Setup}(1^k, \mathcal{U}) \rightarrow (\text{PK}, \text{MK})$  を実行し, システム公開鍵  $\text{PK}$  を  $\mathcal{A}$  に渡す.

**Query Phase:**  $\mathcal{A}$  は以下の各オラクルに対してクエリを任意回数発行することができる.

**秘密鍵生成:**  $\mathcal{A}$  は属性集合  $S_s \in (2^{\mathcal{U}} - \{\phi\})$  をクエリする. 但し, Forgery フェーズで指定するターゲット署名者アクセス構造  $\mathbb{A}_s^*$  に関して  $S_s \in \mathbb{A}_s^*$  を満たす  $S_s$  のクエリは禁止.  $\mathcal{CH}$  は  $\text{SS.KeyGen}(\text{PK}, \text{MK}, S_s) \rightarrow \text{SK}_s$  を計算し  $\mathcal{A}$  へ送る.

**署名生成:**  $\mathcal{A}$  は平文  $m \in \mathcal{M}$ , 属性集合  $S_s \in (2^{\mathcal{U}} - \{\phi\})$ , 署名者アクセス構造  $\mathbb{A}_s (s.t. S_s \in \mathbb{A}_s)$  を  $\mathcal{CH}$  へ送る.  $\mathcal{CH}$  は  $\text{SS.KeyGen}(\text{PK}, \text{MK}, S_s) \rightarrow \text{SK}_s$ ,  $\text{SS.Sig}(\text{PK}, m, \text{SK}_s, \mathbb{A}_s) \rightarrow \sigma$  を実行し  $\mathcal{A}$  へ  $\sigma$  を送る.  $\mathcal{CH}$  は  $(m, \sigma, \mathbb{A}_s)$  を  $\mathcal{L}_{\text{SS.Sig}}$  へ追加する.

**Forgery Phase:**  $\mathcal{A}$  が  $(m^*, \sigma^*, \mathbb{A}_s^*)$  を出力.

SP-ABS 方式  $\Pi_{\text{SS}}$  に関する AP-sEUF-CMA 安全性ゲームにおいて攻撃者  $\mathcal{A}$  の優位性を次式で定義する.

$$\text{Adv}_{\Pi_{\text{SS}}, \mathcal{A}}^{\text{AP-sEUF-CMA}} = \Pr[[m^* \in \mathcal{M}] \wedge [\text{SS.Ver}(\text{PK}, \sigma^*, m^*, \mathbb{A}_s^*) = 1] \wedge [(m^*, \sigma^*, \mathbb{A}_s^*) \notin \mathcal{L}_{\text{SS.Sig}}]] \quad (2.21)$$

**定義 2.15.** 全ての PPTA 攻撃者  $\mathcal{A}$  に対して, SP-ABS 方式  $\Pi_{\text{SS}}$  に関する AP-sEUF-CMA 安全性ゲームにおける  $\mathcal{A}$  の優位性  $\text{Adv}_{\Pi_{\text{SS}}, \mathcal{A}}^{\text{AP-sEUF-CMA}}$  が, セキュリティパラメータ  $k$  に関して無視できるほど小さい値であるならば, SP-ABS 方式  $\Pi_{\text{SS}}$  は AP-sEUF-CMA 安全である.

定義 2.15 は, “強偽造不可能性” である. それに対して, “弱偽造不可能性”, 正式には, 適応的述語モデルにおける適応的選択文書攻撃に対する署名弱偽造不可能性 (Weakly Existentially Unforgeability of Signature under Adaptively Chosen Message Attacks in the Adaptive Predicate model, AP-wEUF-CMA) は, 先の AP-sEUF-CMA 安全性ゲームと同一の AP-wEUF-CMA 安全性ゲームにおいて, 攻撃者の優位性を次式で定義する.

$$\text{Adv}_{\Pi_{\text{SS}}, \mathcal{A}}^{\text{AP-wEUF-CMA}} = \Pr[[m^* \in \mathcal{M}] \wedge [\text{SS.Ver}(\text{PK}, \sigma^*, m^*, \mathbb{A}_s^*) = 1] \wedge [(m^*, \mathbb{A}_s^*) \notin \mathcal{L}_{\text{SS.Sig}}]] \quad (2.22)$$

**定義 2.16.** 全ての PPTA 攻撃者  $\mathcal{A}$  に対して, SP-ABS 方式  $\Pi_{\text{SS}}$  に関する AP-wEUF-CMA 安全性ゲームにおける  $\mathcal{A}$  の優位性  $\text{Adv}_{\Pi_{\text{SS}}, \mathcal{A}}^{\text{AP-wEUF-CMA}}$  が, セキュリティパラメータ  $k$  に関して無視できるほど小さい値であるならば, SP-ABS 方式  $\Pi_{\text{SS}}$  は AP-wEUF-CMA 安全である.

## 2.6.2 SP-sEUF-CMA 安全性

選択的述語モデルにおける適応的選択文書攻撃に対する署名強偽造不可能性 (Strongly Existentially Unforgeability of Signature under Adaptively Chosen Message Attacks in the Selective Predicate model, SP-sEUF-CMA) は, SP-ABS 方式が満たすべき完全性に関する安全性の中で, AP-sEUF-CMA(2.6.1 項) よりも弱いことが自明な安全性である. SP-ABS 方式  $\Pi_{SS}$  に関する SP-sEUF-CMA 安全性は, 攻撃者  $\mathcal{A}$  と挑戦者  $\mathcal{CH}$  との間で行われる以下の SP-sEUF-CMA 安全性ゲームによって定義する.

**Init Phase:**  $\mathcal{CH}$  はターゲット署名者アクセス構造  $\mathbb{A}_s^*$  を  $\mathcal{CH}$  へ送る.

**Setup Phase:**  $\mathcal{CH}$  が  $SS.Setup(1^k, \mathcal{U}) \rightarrow (PK, MK)$  を実行し, システム公開鍵  $PK$  を  $\mathcal{A}$  に渡す.

**Query Phase:**  $\mathcal{A}$  は以下の各オラクルに対してクエリを任意回数発行することができる.

**秘密鍵生成:**  $\mathcal{A}$  は属性集合  $S_s \in (2^{\mathcal{U}} - \{\phi\})$  をクエリする. 但し,  $S_s \in \mathbb{A}_s^*$  を満たす  $S_s$  のクエリは禁止.  $\mathcal{CH}$  は  $SS.KeyGen(PK, MK, S_s) \rightarrow SK_s$  を計算し  $\mathcal{A}$  へ送る.

**署名生成:**  $\mathcal{A}$  は平文  $m \in \mathcal{M}$ , 属性集合  $S_s \in (2^{\mathcal{U}} - \{\phi\})$ , 署名者アクセス構造  $\mathbb{A}_s (s.t. S_s \in \mathbb{A}_s)$  を  $\mathcal{CH}$  へ送る.  $\mathcal{CH}$  は  $SS.KeyGen(PK, MK, S_s) \rightarrow SK_s$ ,  $SS.Sig(PK, m, SK_s, \mathbb{A}_s) \rightarrow \sigma$  を実行し  $\mathcal{A}$  へ  $\sigma$  を送る.  $\mathcal{CH}$  は  $(m, \sigma, \mathbb{A}_s)$  を  $\mathcal{L}_{SS.Sig}$  へ追加する.

**Forgery Phase:**  $\mathcal{A}$  が  $(m^*, \sigma^*)$  を出力.

SP-ABS 方式  $\Pi_{SS}$  に関する SP-sEUF-CMA 安全性ゲームにおいて攻撃者  $\mathcal{A}$  の優位性を次式で定義する.

$$\text{Adv}_{\Pi_{SS}, \mathcal{A}}^{\text{SP-sEUF-CMA}} = \Pr[[m^* \in \mathcal{M}] \wedge [SS.Ver(PK, \sigma^*, m^*, \mathbb{A}_s^*) = 1] \wedge [(m^*, \sigma^*, \mathbb{A}_s^*) \notin \mathcal{L}_{SS.Sig}]] \quad (2.23)$$

**定義 2.17.** 全ての PPTA 攻撃者  $\mathcal{A}$  に対して, SP-ABS 方式  $\Pi_{SS}$  に関する SP-sEUF-CMA 安全性ゲームにおける  $\mathcal{A}$  の優位性  $\text{Adv}_{\Pi_{SS}, \mathcal{A}}^{\text{SP-sEUF-CMA}}$  が, セキュリティパラメータ  $k$  に関して無視できるほど小さい値であるならば, SP-ABS 方式  $\Pi_{SS}$  は SP-sEUF-CMA 安全である.

定義 2.17 は, “強偽造不可能性” である. それに対して, “弱偽造不可能性”, 正式には, 選択的述語モデルにおける適応的選択文書攻撃に対する署名弱偽造不可能性 (Weakly

Existentially Unforgeability of Signature under Adaptively Chosen Message Attacks in the Selective Predicate model, SP-wEUF-CMA) は、先の SP-sEUF-CMA 安全性ゲームと同一の SP-wEUF-CMA 安全性ゲームにおいて、攻撃者の優位性を次式で定義する。

$$\text{Adv}_{\Pi_{\text{SS}}, \mathcal{A}}^{\text{SP-wEUF-CMA}} = \Pr[[m^* \in \mathcal{M}] \wedge [\text{SS.Ver}(\text{PK}, \sigma^*, m^*, \mathbb{A}_s) = 1] \wedge [(m^*, \mathbb{A}_s) \notin \mathcal{L}_{\text{SS.Sig}}]] \quad (2.24)$$

**定義 2.18.** 全ての PPTA 攻撃者  $\mathcal{A}$  に対して、SP-ABS 方式  $\Pi_{\text{SS}}$  に関する SP-wEUF-CMA 安全性ゲームにおける  $\mathcal{A}$  の優位性  $\text{Adv}_{\Pi_{\text{SS}}, \mathcal{A}}^{\text{SP-wEUF-CMA}}$  が、セキュリティパラメータ  $k$  に関して無視できるほど小さい値であるならば、SP-ABS 方式  $\Pi_{\text{SS}}$  は SP-wEUF-CMA 安全である。

### 2.6.3 完全匿名性

SP-ABS 方式の完全匿名性 (Perfect Privacy, あるいは Signer Privacy) は、直感的な説明としては、署名から署名者の属性集合に関する情報が漏れないことを保障する安全性である。具体的には、次のように定義される。

**定義 2.19.** SP-ABS 方式が完全匿名性を満たすとは、当該方式が以下の条件を満たす場合を言う。：

全ての  $(\text{PK}, \text{MK}) \leftarrow \text{SS.Setup}(1^k, \mathcal{U})$ , 全ての  $S_s \in (2^u - \{\phi\})$ , 全ての  $S'_s \in (2^u - \{\phi\})$ , 全ての  $\text{SK}_s \leftarrow \text{SS.KeyGen}(\text{PK}, \text{MK}, S_s)$ , 全ての  $\text{SK}'_s \leftarrow \text{SS.KeyGen}(\text{PK}, \text{MK}, S'_s)$ , 全ての  $m \in \mathcal{M}$ , 全ての  $\mathbb{A}_s (s.t. S_s \in \mathbb{A}_s \wedge S'_s \in \mathbb{A}_s)$  に対して、 $\text{SS.Sig}(\text{PK}, m, \text{SK}_s, \mathbb{A}_s)$  の確率分布と  $\text{SS.Sig}(\text{PK}, m, \text{SK}'_s, \mathbb{A}_s)$  の確率分布が同一である。

### 2.6.4 署名者アクセス構造衝突困難性

SP-ABS 方式の署名者アクセス構造衝突困難性は著者が独自に命名し、かつ定義した性質 (安全性) である。

SP-ABS 方式  $\Pi_{\text{SS}}$  に関する署名者アクセス構造衝突困難性は、攻撃者  $\mathcal{A}$  と挑戦者  $\mathcal{CH}$  との間で行われる以下の署名者アクセス構造衝突困難性ゲームによって定義される。

**Setup Phase:**  $\mathcal{CH}$  が  $\text{SS.Setup}(1^k, \mathcal{U}) \rightarrow (\text{PK}, \text{MK})$  を実行し、システム公開鍵  $\text{PK}$  を  $\mathcal{A}$  に渡す。

**Query Phase 1:**  $\mathcal{A}$  は以下の各オラクルに対してクエリを任意回数発行することができる。

**秘密鍵生成:**  $\mathcal{A}$  は属性集合  $S_s \in (2^u - \{\phi\})$  をクエリする。 $\mathcal{CH}$  は  $\text{SS.KeyGen}(\text{PK}, \text{MK}, S_s) \rightarrow \text{SK}_s$  を計算し  $\mathcal{A}$  へ送る。

**Challenge Phase:**  $\mathcal{A}$  は平文  $m^*$ , ターゲット署名者アクセス構造  $A_s^*, S_s^* \in \mathbb{A}_s^*$  を満たす属性集合  $S_s^*$  を送る.  $\mathcal{CH}$  は  $\text{SS.KeyGen}(\text{PK}, \text{MK}, S_s^*) \rightarrow \text{SK}_s^*$ ,  $\text{SS.Sig}(\text{PK}, m^*, \text{SK}_s^*, A_s^*) \rightarrow \sigma^*$  を実行する. 最後に  $\mathcal{CH}$  は  $\mathcal{A}$  へ  $\sigma^*$  を送る.

**Query Phase 2**  $\mathcal{A}$  は以下の各オラクルに対してクエリを任意回数発行することができる.

**秘密鍵生成:** Query Phase 1 と同じ.

**Output Phase:**  $\mathcal{A}$  が  $A'_s$  を出力.

SP-ABS 方式  $\Pi_{\text{SS}}$  に関する署名者アクセス構造衝突困難性ゲームにおいて攻撃者  $\mathcal{A}$  の優位性を次式で定義する.

$$\text{Adv}_{\Pi_{\text{SS}}, \mathcal{A}}^{\text{SASCR}} = \Pr[A'_s \neq A_s^* \wedge \text{SS.Ver}(\text{PK}, \sigma^*, m^*, A'_s) = 1] \quad (2.25)$$

**定義 2.20.** 全ての  $PPTA$  攻撃者  $\mathcal{A}$  に対して,  $SP\text{-ABS}$  方式  $\Pi_{\text{SS}}$  に関する署名者アクセス構造衝突困難性ゲームにおける  $\mathcal{A}$  の優位性  $\text{Adv}_{\Pi_{\text{SS}}, \mathcal{A}}^{\text{SASCR}}$  が, セキュリティパラメータ  $k$  に関して無視できるほど小さい値であるならば,  $SP\text{-ABS}$  方式  $\Pi_{\text{SS}}$  は署名者アクセス構造衝突困難性を備える.

署名者アクセス構造衝突困難性は, 本稿で独自に定義した性質であり, 既存研究で既存方式が当該性質を満たすことの証明はなされておらず, かつ, 本稿でもその証明は完成していない. しかし, 直感的には当該性質は極めて自然な性質であると考えられる. そのように考えられる根拠等については, 8.2.2 項で論じる.

## 2.7 鍵ポリシー型属性ベース署名

鍵ポリシー型属性ベース署名 (Key-Policy Attribute-Based Key Encapsulation Mechanism, 以降略称として “KP-ABS” と表記することもある) は, 秘密鍵とポリシー (アクセス構造), 署名と属性集合 (署名者属性集合) がそれぞれ対応する, 属性ベース署名である.

SP-ABS は以下の4つのアルゴリズムから構成される.

$\text{KS.Setup}(1^k, \mathcal{U}) \rightarrow (\text{PK}, \text{MK})$

セットアップアルゴリズム  $\text{KS.Setup}$  は, セキュリティパラメータ  $k$  と, 属性の全体集合  $\mathcal{U}$  を入力とし, システム公開鍵  $\text{PK}$  とマスター秘密鍵  $\text{MK}$  を出力する.

$\text{KS.KeyGen}(\text{PK}, \text{MK}, A_s) \rightarrow \text{SK}_s$

鍵生成アルゴリズム  $\text{KS.KeyGen}$  は, システム公開鍵  $\text{PK}$ , マスター秘密鍵  $\text{MK}$ , 送信者のポリシー  $A_s \in (2^{\mathcal{U}} - \{\emptyset\})$  を入力とし, 送信者用秘密鍵  $\text{SK}_s$  を出力する.

$\text{KS.Sig}(\text{PK}, m, \text{SK}_s, S_s) \rightarrow \sigma$

署名生成アルゴリズム  $\text{KS.Sig}$  は、システム公開鍵  $\text{PK}$ 、平文  $m \in \mathcal{M}$ 、送信者用秘密鍵  $\text{SK}_s$ 、署名者属性集合  $S_s$  を入力とし、署名  $\sigma$  を出力する。

$\text{KS.Ver}(\text{PK}, \sigma, m, \mathbb{A}_s) \rightarrow 1 / 0$

署名検証アルゴリズム  $\text{KS.Ver}$  は、システム公開鍵  $\text{PK}$ 、署名  $\sigma$ 、平文  $m \in \mathcal{M}$ 、署名者属性集合  $S_s$  を入力とし、署名検証合格を表す“1”、あるいは不合格を表す“0”を出力する。

**正当性 (Correctness)**  $\text{KP-ABS}$  方式は正当であることが求められる。 $\text{KP-ABS}$  方式が正当であるとは、全ての  $k$ 、全ての  $\mathcal{U}$ 、全ての  $(\text{PK}, \text{MK}) \leftarrow \text{KS.Setup}(1^k, \mathcal{U})$ 、全ての  $m \in \mathcal{M}$ 、全ての  $\mathbb{A}_s$ 、全ての  $\text{SK}_s \leftarrow \text{KS.KeyGen}(\text{PK}, \text{MK}, \mathbb{A}_s)$ 、 $S_s \in \mathbb{A}_s$  を満たす全ての  $S_s$ 、全ての  $\sigma \leftarrow \text{KS.Sig}(\text{PK}, m, \text{SK}_s, \mathbb{A}_s)$  に対して、次の等式が成立する場合を言う。

$$\Pr[\text{KS.Ver}(\text{PK}, \sigma, m, \mathbb{A}_s) = 1] = 1 \quad (2.26)$$

## 2.7.1 AA-sEUF-CMA 安全性

適応的的属性モデルにおける適応的選択文書攻撃に対する署名強偽造不可能性 (Strongly Existentially Unforgeability of Signature under Adaptively Chosen Message Attacks in the Adaptive Attribute Model, AA-sEUF-CMA) は、 $\text{KP-ABS}$  方式が満たすべき完全性に関する安全性の中で、最も強い安全性である。 $\text{KP-ABS}$  方式  $\Pi_{\text{KS}}$  に関する AA-sEUF-CMA 安全性は、攻撃者  $\mathcal{A}$  と挑戦者  $\text{CH}$  との間で行われる以下の AA-sEUF-CMA 安全性ゲームによって定義する。

**Setup Phase:**  $\text{CH}$  が  $\text{KS.Setup}(1^k, \mathcal{U}) \rightarrow (\text{PK}, \text{MK})$  を実行し、システム公開鍵  $\text{PK}$  を  $\mathcal{A}$  に渡す。

**Query Phase:**  $\mathcal{A}$  は以下の各オラクルに対してクエリを任意回数発行することができる。

**秘密鍵生成:**  $\mathcal{A}$  はポリシー  $\mathbb{A}_s$  をクエリする。但し、Forgery フェーズで指定するターゲット署名者属性集合  $S_s^*$  に関して  $S_s^* \in \mathbb{A}_s$  を満たす  $\mathbb{A}_s$  のクエリは禁止。 $\text{CH}$  は  $\text{KS.KeyGen}(\text{PK}, \text{MK}, \mathbb{A}_s) \rightarrow \text{SK}_s$  を計算し  $\mathcal{A}$  へ送る。

**署名生成:**  $\mathcal{A}$  は平文  $m \in \mathcal{M}$ 、ポリシー  $\mathbb{A}_s$ 、署名者属性集合  $S_s (s.t. S_s \in \mathbb{A}_s)$  を  $\text{CH}$  へ送る。 $\text{CH}$  は  $\text{KS.KeyGen}(\text{PK}, \text{MK}, \mathbb{A}_s) \rightarrow \text{SK}_s$ 、 $\text{KS.Sig}(\text{PK}, m, \text{SK}_s, S_s) \rightarrow \sigma$  を実行し  $\mathcal{A}$  へ  $\sigma$  を送る。 $\text{CH}$  は  $(m, \sigma, S_s)$  を  $\mathcal{L}_{\text{KS.Sig}}$  へ追加する。

**Forgery Phase:**  $\mathcal{A}$  が  $(m^*, \sigma^*, S_s^*)$  を出力。

KP-ABS 方式  $\Pi_{KS}$  に関する AA-sEUF-CMA 安全性ゲームにおいて攻撃者  $\mathcal{A}$  の優位性を次式で定義する.

$$\text{Adv}_{\Pi_{KS}, \mathcal{A}}^{\text{AA-sEUF-CMA}} = \Pr[[m^* \in \mathcal{M}] \wedge [\text{KS.Ver}(\text{PK}, \sigma^*, m^*, S_s^*) = 1] \wedge [(m^*, \sigma^*, S_s^*) \notin \mathcal{L}_{\text{KS.Sig}}]] \quad (2.27)$$

**定義 2.21.** 全ての PPTA 攻撃者  $\mathcal{A}$  に対して, KP-ABS 方式  $\Pi_{KS}$  に関する AA-sEUF-CMA 安全性ゲームにおける  $\mathcal{A}$  の優位性  $\text{Adv}_{\Pi_{KS}, \mathcal{A}}^{\text{AA-sEUF-CMA}}$  が, セキュリティパラメータ  $k$  に関して無視できるほど小さい値であるならば, KP-ABS 方式  $\Pi_{KS}$  は AA-sEUF-CMA 安全である.

定義 2.21 は, “強偽造不可能性”である. それに対して, “弱偽造不可能性”, 正式には, 適応的的属性モデルにおける適応的選択文書攻撃に対する署名弱偽造不可能性 (Weakly Existentially Unforgeability of Signature under Adaptively Chosen Message Attacks in the Adaptive Attribute model, AA-wEUF-CMA) は, 先の AA-sEUF-CMA 安全性ゲームと同一の AA-wEUF-CMA 安全性ゲームにおいて, 攻撃者の優位性を次式で定義する.

$$\text{Adv}_{\Pi_{KS}, \mathcal{A}}^{\text{AA-wEUF-CMA}} = \Pr[[m^* \in \mathcal{M}] \wedge [\text{KS.Ver}(\text{PK}, \sigma^*, m^*, S_s^*) = 1] \wedge [(m^*, S_s^*) \notin \mathcal{L}_{\text{KS.Sig}}]] \quad (2.28)$$

**定義 2.22.** 全ての PPTA 攻撃者  $\mathcal{A}$  に対して, KP-ABS 方式  $\Pi_{KS}$  に関する AA-wEUF-CMA 安全性ゲームにおける  $\mathcal{A}$  の優位性  $\text{Adv}_{\Pi_{KS}, \mathcal{A}}^{\text{AA-wEUF-CMA}}$  が, セキュリティパラメータ  $k$  に関して無視できるほど小さい値であるならば, KP-ABS 方式  $\Pi_{KS}$  は AA-wEUF-CMA 安全である.

## 2.7.2 SA-sEUF-CMA 安全性

選択的的属性モデルにおける適応的選択文書攻撃に対する署名強偽造不可能性 (Strongly Existentially Unforgeability of Signature under Adaptively Chosen Message Attacks in the Selective Attribute Model, SA-sEUF-CMA) は, KP-ABS 方式が満たすべき完全性に関する安全性の中で, AA-sEUF-CMA(2.7.1 項) よりも弱いことが自明な安全性である. KP-ABS 方式  $\Pi_{KS}$  に関する SA-sEUF-CMA 安全性は, 攻撃者  $\mathcal{A}$  と挑戦者  $\mathcal{CH}$  との間で行われる以下の SA-sEUF-CMA 安全性ゲームによって定義する.

**Init Phase:**  $\mathcal{CH}$  はターゲット署名者属性集合  $S_s^*$  を  $\mathcal{CH}$  へ送る.

**Setup Phase:**  $\mathcal{CH}$  が  $\text{KS.Setup}(1^k, \mathcal{U}) \rightarrow (\text{PK}, \text{MK})$  を実行し, システム公開鍵  $\text{PK}$  を  $\mathcal{A}$  に渡す.

**Query Phase:**  $\mathcal{A}$  は以下の各オラクルに対してクエリを任意回数発行すること

ができる.

**秘密鍵生成:**  $\mathcal{A}$  はポリシー  $\mathbb{A}_s$  をクエリする. 但し, **Forgery** フェーズで指定するターゲット署名者属性集合  $S_s^*$  に関して  $S_s^* \in \mathbb{A}_s$  を満たす  $\mathbb{A}_s$  のクエリは禁止.  $\mathcal{CH}$  は  $\text{KS.KeyGen}(\text{PK}, \text{MK}, \mathbb{A}_s) \rightarrow \text{SK}_s$  を計算し  $\mathcal{A}$  へ送る.

**署名生成:**  $\mathcal{A}$  は平文  $m \in \mathcal{M}$ , ポリシー  $\mathbb{A}_s$ , 署名者属性集合  $S_s (s.t. S_s \in \mathbb{A}_s)$  を  $\mathcal{CH}$  へ送る.  $\mathcal{CH}$  は  $\text{KS.KeyGen}(\text{PK}, \text{MK}, \mathbb{A}_s) \rightarrow \text{SK}_s$ ,  $\text{KS.Sig}(\text{PK}, m, \text{SK}_s, S_s) \rightarrow \sigma$  を実行し  $\mathcal{A}$  へ  $\sigma$  を送る.  $\mathcal{CH}$  は  $(m, \sigma, S_s)$  を  $\mathcal{L}_{\text{KS.Sig}}$  へ追加する.

**Forgery Phase:**  $\mathcal{A}$  が  $(m^*, \sigma^*)$  を出力.

KP-ABS 方式  $\Pi_{\text{KS}}$  に関する SA-sEUF-CMA 安全性ゲームにおいて攻撃者  $\mathcal{A}$  の優位性を次式で定義する.

$$\text{Adv}_{\Pi_{\text{KS}}, \mathcal{A}}^{\text{SA-sEUF-CMA}} = \Pr[[m^* \in \mathcal{M}] \wedge [\text{KS.Ver}(\text{PK}, \sigma^*, m^*, S_s^*) = 1] \wedge [(m^*, \sigma^*, S_s^*) \notin \mathcal{L}_{\text{KS.Sig}}]] \quad (2.29)$$

**定義 2.23.** 全ての PPTA 攻撃者  $\mathcal{A}$  に対して, KP-ABS 方式  $\Pi_{\text{KS}}$  に関する SA-sEUF-CMA 安全性ゲームにおける  $\mathcal{A}$  の優位性  $\text{Adv}_{\Pi_{\text{KS}}, \mathcal{A}}^{\text{SA-sEUF-CMA}}$  が, セキュリティパラメータ  $k$  に関して無視できるほど小さい値であるならば, KP-ABS 方式  $\Pi_{\text{KS}}$  は SA-sEUF-CMA 安全である.

定義 2.23 は, “強偽造不可能性” である. それに対して, “弱偽造不可能性”, 正式には, 選択的属性モデルにおける適応的選択文書攻撃に対する署名弱偽造不可能性 (Weakly Existentially Unforgeability of Signature under Adaptively Chosen Message Attacks in the Selective Attribute model, SA-wEUF-CMA) は, 先の SA-sEUF-CMA 安全性ゲームと同一の SA-wEUF-CMA 安全性ゲームにおいて, 攻撃者の優位性を次式で定義する.

$$\text{Adv}_{\Pi_{\text{KS}}, \mathcal{A}}^{\text{SA-wEUF-CMA}} = \Pr[[m^* \in \mathcal{M}] \wedge [\text{KS.Ver}(\text{PK}, \sigma^*, m^*, S_s^*) = 1] \wedge [(m^*, S_s^*) \notin \mathcal{L}_{\text{KS.Sig}}]] \quad (2.30)$$

**定義 2.24.** 全ての PPTA 攻撃者  $\mathcal{A}$  に対して, KP-ABS 方式  $\Pi_{\text{KS}}$  に関する SA-wEUF-CMA 安全性ゲームにおける  $\mathcal{A}$  の優位性  $\text{Adv}_{\Pi_{\text{KS}}, \mathcal{A}}^{\text{SA-wEUF-CMA}}$  が, セキュリティパラメータ  $k$  に関して無視できるほど小さい値であるならば, KP-ABS 方式  $\Pi_{\text{KS}}$  は SA-wEUF-CMA 安全である.

### 2.7.3 完全匿名性

KP-ABS 方式の完全匿名性 (Perfect Privacy, あるいは Signer Privacy) は, 直感的な説明としては, 署名から署名者のポリシーに関する情報が漏れないことを保障する安全性である. 具体的には, 次のように定義される.

**定義 2.25.** KP-ABS 方式が完全匿名性を満たすとは, 当該方式が以下の条件を満たす場合を言う. :

全ての  $(PK, MK) \leftarrow \text{KS.Setup}(1^k, \mathcal{U})$ , 全ての  $A_s$ , 全ての  $A'_s$ , 全ての  $SK_s \leftarrow \text{KS.KeyGen}(PK, MK, A_s)$ , 全ての  $SK'_s \leftarrow \text{KS.KeyGen}(PK, MK, A'_s)$ , 全ての  $m \in \mathcal{M}$ , 全ての  $S_s$  ( $s.t. S_s \in A_s \wedge S_s \in A'_s$ ) に対して,  $\text{KS.Sig}(PK, m, SK_s, S_s)$  の確率分布と  $\text{KS.Sig}(PK, m, SK'_s, S_s)$  の確率分布が同一である.

### 2.7.4 署名者属性集合衝突困難性

KP-ABS 方式の署名者属性集合衝突困難性は著者が独自に命名し, かつ定義した性質 (安全性) である.

KP-ABS 方式  $\Pi_{\text{KS}}$  に関する署名者属性集合衝突困難性は, 攻撃者  $\mathcal{A}$  と挑戦者  $\mathcal{CH}$  との間で行われる以下の署名者属性集合衝突困難性ゲームによって定義される.

**Setup Phase:**  $\mathcal{CH}$  が  $\text{KS.Setup}(1^k, \mathcal{U}) \rightarrow (PK, MK)$  を実行し, システム公開鍵  $PK$  を  $\mathcal{A}$  に渡す.

**Query Phase 1:**  $\mathcal{A}$  は以下の各オラクルに対してクエリを任意回数発行することができる.

**秘密鍵生成:**  $\mathcal{A}$  はアクセス構造  $A_s$  をクエリする.  $\mathcal{CH}$  は  $\text{KS.KeyGen}(PK, MK, A_s) \rightarrow SK_s$  を計算し  $\mathcal{A}$  へ送る.

**Challenge Phase:**  $\mathcal{A}$  は平文  $m^*$ , ターゲット署名者属性集合  $S_s^*$ ,  $S'_s^* \in A_s^*$  を満たすアクセス構造  $A_s^*$  を送る.  $\mathcal{CH}$  は  $\text{KS.KeyGen}(PK, MK, A_s^*) \rightarrow SK_s^*$ ,  $\text{KS.Sig}(PK, m^*, SK_s^*, S_s^*) \rightarrow \sigma^*$  を実行する. 最後に  $\mathcal{CH}$  は  $\mathcal{A}$  へ  $\sigma^*$  を送る.

**Query Phase 2**  $\mathcal{A}$  は以下の各オラクルに対してクエリを任意回数発行することができる.

**秘密鍵生成:** Query Phase 1 と同じ.

**Output Phase:**  $\mathcal{A}$  が  $S'_s$  を出力.

KP-ABS 方式  $\Pi_{\text{KS}}$  に関する署名者属性集合衝突困難性ゲームにおいて攻撃者  $\mathcal{A}$  の

優位性を次式で定義する.

$$\text{Adv}_{\Pi_{\text{KS}}, \mathcal{A}}^{\text{SASC}R} = \Pr[S'_s \neq S_s^* \wedge \text{KS.Ver}(\text{PK}, \sigma^*, m^*, S'_s) = 1] \quad (2.31)$$

**定義 2.26.** 全ての  $PPTA$  攻撃者  $\mathcal{A}$  に対して,  $KP\text{-ABS}$  方式  $\Pi_{\text{KS}}$  に関する署名者属性集合衝突困難性ゲームにおける  $\mathcal{A}$  の優位性  $\text{Adv}_{\Pi_{\text{KS}}, \mathcal{A}}^{\text{SASC}R}$  が, セキュリティパラメータ  $k$  に関して無視できるほど小さい値であるならば,  $KP\text{-ABS}$  方式  $\Pi_{\text{KS}}$  は署名者属性集合衝突困難性を備える.

署名者属性集合衝突困難性は, 本稿で独自に定義した性質であり, 既存研究で既存方式が当該性質を満たすことの証明はなされておらず, かつ, 本稿でもその証明は完成していない. しかし, 直感的には当該性質は極めて自然な性質であると考えられる. そのように考えられる根拠等については, 8.2.2 項で論じる.

## 2.8 暗号文ポリシー型属性ベース Signcryption

暗号文ポリシー型属性ベース Signcryption (Ciphertext-Policy Attribute-Based Signcryption, 以降略称として“CP-ABSC”と表記することもある) は, CP-ABE と SP-ABS の両機能を同時に実現することが可能な高機能公開鍵暗号である. 具体的には, サインクリプタはある平文に関して自身の属性集合が満たす様な署名者アクセス構造を指定し, 同時に適切な復号者アクセス構造を指定し, 自身の秘密鍵を使って平文に対してサインクリプション処理を行い, サインクリプテキストを生成する. そして, 復号者アクセス集合を満たす様な属性集合を持つアンサインクリプタは, 自身の秘密鍵を使ってサインクリプテキストに対してアンサインクリプション処理を行い, 平文を入手し, かつ「署名者アクセス構造を満たす様な属性集合を持つ人物が確実にその平文を作成した事実」を確認することができる.

CP-ABSC 方式は以下の 5 つのアルゴリズムから構成される.

$\text{CS.Setup}(1^k, \mathcal{U}_s, \mathcal{U}_r) \rightarrow (\text{PK}, \text{MK})$

セットアップアルゴリズム  $\text{CS.Setup} \rightarrow (\text{PK}, \text{MK})$  は, セキュリティパラメータ  $k$ , 送信者の属性の全体集合  $\mathcal{U}_s$ , 受信者の属性の全体集合  $\mathcal{U}_r$  を入力とし, システム公開鍵  $\text{PK}$  とマスター秘密鍵  $\text{MK}$  を出力する.

$\text{CS.KeyGen}_s(\text{PK}, \text{MK}, S_s) \rightarrow \text{SK}_s$

送信者用秘密鍵生成アルゴリズム  $\text{CS.KeyGen}_s$  は, システム公開鍵  $\text{PK}$ , マスター秘密鍵  $\text{MK}$ , 送信者の属性集合  $S_s \in (2^{\mathcal{U}_s} - \{\emptyset\})$  を入力とし, 送信者用秘密鍵  $\text{SK}_s$  を出力する.

$\text{CS.KeyGen}_r(\text{PK}, \text{MK}, S_r) \rightarrow \text{SK}_r$

受信者用秘密鍵生成アルゴリズム  $\text{CS.KeyGen}_r$  は, システム公開鍵  $\text{PK}$ , マスター秘密鍵  $\text{MK}$ , 受信者の属性集合  $S_r \in (2^{\mathcal{U}_r} - \{\emptyset\})$  を入力とし, 受信者用秘密鍵  $\text{SK}_r$  を出力する.

$\text{CS.SC}(\text{PK}, m, \text{SK}_s, \mathbb{A}_s, \mathbb{A}_d) \rightarrow C$

サインクリプションアルゴリズム  $\text{CS.SC}$  は、システム公開鍵  $\text{PK}$ 、平文  $m \in \mathcal{M}$ 、送信者用秘密鍵  $\text{SK}_s$ 、署名者アクセス構造  $\mathbb{A}_s$ 、復号者アクセス構造  $\mathbb{A}_d$  を入力とし、サインクリプトテキスト  $C \in \mathcal{C}$  を出力する。

$\text{CS.USC}(\text{PK}, C, \text{SK}_r, \mathbb{A}_s) \rightarrow m / \perp$

アンサインクリプションアルゴリズム  $\text{CS.USC}$  は、システム公開鍵  $\text{PK}$ 、サインクリプトテキスト  $C \in \mathcal{C}$ 、署名者アクセス構造  $\mathbb{A}_s$ 、受信者用秘密鍵  $\text{SK}_r$  を入力とし、平文  $m \in \mathcal{M}$  または  $\perp$  を出力する。

**正当性 (Correctness)** CP-ABSC 方式は正当であることが求められる。CP-ABSC 方式が正当であるとは、全ての  $k$ 、全ての  $\mathcal{U}_s$ 、全ての  $\mathcal{U}_r$ 、全ての  $(\text{PK}, \text{MK}) \leftarrow \text{CS.Setup}(1^k, \mathcal{U}_s, \mathcal{U}_r)$ 、全ての  $S_s \in (2^{\mathcal{U}_s} - \{\phi\})$ 、全ての  $\text{SK}_s \leftarrow \text{CS.KeyGen}_S(\text{PK}, \text{MK}, S_s)$ 、 $S_s \in \mathbb{A}_s$  を満たす全ての  $\mathbb{A}_s$ 、全ての  $S_r \in (2^{\mathcal{U}_r} - \{\phi\})$ 、全ての  $\text{SK}_r \leftarrow \text{CS.KeyGen}_R(\text{PK}, \text{MK}, S_r)$ 、 $S_r \in \mathbb{A}_d$  を満たす全ての  $S_r$ 、全ての  $m \in \mathcal{M}$ 、全ての  $C \leftarrow \text{CS.SC}(\text{PK}, m, \text{SK}_s, \mathbb{A}_s, \mathbb{A}_d)$  に対して、次の等式が成立する場合を言う。

$$\Pr[\text{CS.USC}(\text{PK}, C, \text{SK}_r, \mathbb{A}_s) = m] = 1 \quad (2.32)$$

## 2.8.1 AP-IND-CCA 安全性

適応的述語モデルにおける適応的選択暗号文(サインクリプトテキスト)攻撃に対する暗号文(サインクリプトテキスト)識別不可能性 (Ciphertext(Signcryptext)-Indistinguishability under Adaptively Chosen Ciphertext(Signcryptext) Attacks in the Adaptive Predicate Model, AP-IND-CCA) は、CP-ABSC 方式が満たすべき秘匿性に関する安全性の中で、最も強いと考えられている安全性である。CP-ABSC 方式  $\Pi_{\text{CS}}$  に関する AP-IND-CCA 安全性は、攻撃者  $\mathcal{A}$  と挑戦者  $\mathcal{CH}$  との間で行われる以下の AP-IND-CCA 安全性ゲームによって定義する。

**Setup Phase:**  $\mathcal{CH}$  が  $\text{CS.Setup}(1^k, \mathcal{U}_s, \mathcal{U}_r) \rightarrow (\text{PK}, \text{MK})$  を実行し、システム公開鍵  $\text{PK}$  を  $\mathcal{A}$  に渡す。

**Query Phase 1:**  $\mathcal{A}$  は以下の各オラクルに対してクエリを任意回数発行することができる

**送信者用鍵生成:**  $\mathcal{A}$  は属性集合  $S_s \in (2^{\mathcal{U}_s} - \{\phi\})$  をクエリする。 $\mathcal{CH}$  は  $\text{CS.KeyGen}_S(\text{PK}, \text{MK}, S_s) \rightarrow \text{SK}_s$  を実行し  $\text{SK}_s$  を  $\mathcal{A}$  へ送る。

**受信者用鍵生成:**  $\mathcal{A}$  は属性集合  $S_r \in (2^{\mathcal{U}_r} - \{\phi\})$  をクエリする。但し、後に Challenge Phase で選択する  $\mathbb{A}_d^*$  に関して、 $S_r \in \mathbb{A}_d^*$  を満たす  $S_r$  のクエリは禁止。 $\mathcal{CH}$  は  $\text{KeyGen}_R(\text{PK}, \text{MK}, S_r) \rightarrow \text{SK}_r$  を実行し  $\text{SK}_r$  を

$\mathcal{A}$  へ送る.

**サインクリプションオラクル:**  $\mathcal{A}$  は平文  $m \in \mathcal{M}$ , 署名者アクセス構造  $\mathbb{A}_s$ , 復号者アクセス構造  $\mathbb{A}_d$ ,  $S_s \in \mathbb{A}_s$  を満たす属性集合  $S_s \in (2^{\mathcal{U}_s} - \{\emptyset\})$  をクエリする.  $\mathcal{CH}$  は  $\text{CS.KeyGen}_s(\text{PK}, \text{MK}, S_s) \rightarrow \text{SK}_s$ ,  $\text{CS.SC}(\text{PK}, m, \text{SK}_s, \mathbb{A}_s, \mathbb{A}_d) \rightarrow C$  を実行し,  $C$  を  $\mathcal{A}$  へ送る.

**アンサインクリプションオラクル:**  $\mathcal{A}$  はサインクリプトテキスト  $C \in \mathcal{C}$ , 署名者アクセス構造  $\mathbb{A}_s$ , 属性集合  $S_r \in (2^{\mathcal{U}_r} - \{\emptyset\})$  をクエリする.  $\mathcal{CH}$  は  $\text{CS.KeyGen}_r(\text{PK}, \text{MK}, S_r) \rightarrow \text{SK}_r$ ,  $\text{CS.USC}(\text{PK}, C, \text{SK}_r, \mathbb{A}_s) \rightarrow m/\perp$  を実行し最終実行結果を  $\mathcal{A}$  へ送る.

**Challenge Phase:**  $\mathcal{A}$  は復号者アクセス構造  $\mathbb{A}_d^*$ , 署名者アクセス構造  $\mathbb{A}_s^*$ , 属性集合  $S_s^* \in \mathbb{A}_s^*$ , 及びサイズの等しい二つの平文  $m_0 \in \mathcal{M}, m_1 \in \mathcal{M}$  を選び  $\mathcal{CH}$  に送る.  $\mathcal{CH}$  は  $\text{SK}_s^* \leftarrow \text{CS.KeyGen}_s(\text{PK}, \text{MK}, S_s^*), b \stackrel{\text{U}}{\leftarrow} \{0, 1\}, C^* \leftarrow \text{CS.SC}(\text{PK}, m_b^*, \text{SK}_s^*, \mathbb{A}_s^*, \mathbb{A}_d^*)$  を実行し, チャレンジサインクリプトテキスト  $C^*$  を  $\mathcal{A}$  に送る.

**Query Phase 2:**  $\mathcal{A}$  は以下の各オラクルに対してクエリを任意回数発行することができる.

**送信者用鍵生成:** Query Phase 1 と同様.

**受信者用鍵生成:** Query Phase 1 と同様. 但し,  $S_r \in \mathbb{A}_d^*$  を満たす  $S_r$  のクエリは禁止.

**サインクリプションオラクル:** Query Phase 1 と同様.

**アンサインクリプションオラクル:** Query Phase 1 と同様. 但し,  $(C, \mathbb{A}_s, S_r) = (C^*, \mathbb{A}_s^*, S_r (s.t. S_r \in \mathbb{A}_d^*))$  のクエリは禁止.

**Guess Phase:**  $\mathcal{A}$  はチャレンジビット  $b$  に対する推測として  $b' \in \{0, 1\}$  を出力.

CP-ABSC 方式  $\Pi_{\text{CS}}$  に関する AP-IND-CCA 安全性ゲームにおいて攻撃者  $\mathcal{A}$  の優位性を次式で定義する.

$$\text{Adv}_{\Pi_{\text{CS}}, \mathcal{A}}^{\text{AP-IND-CCA}} = |\Pr[b' = b] - \frac{1}{2}| \quad (2.33)$$

**定義 2.27.** 全ての PPTA 攻撃者  $\mathcal{A}$  に対して, CP-ABSC 方式  $\Pi_{\text{CS}}$  に関する AP-IND-CCA 安全性ゲームにおける  $\mathcal{A}$  の優位性  $\text{Adv}_{\Pi_{\text{CS}}, \mathcal{A}}^{\text{AP-IND-CCA}}$  が, セキュリティパラメータ  $k$  に関して無視できるほど小さい値であるならば, CP-ABSC 方式  $\Pi_{\text{CS}}$  は AP-IND-CCA 安全である.

AP-IND-CCA 安全性は“内部者安全性”である.

本パラグラフでは, CP-ABSC の AP-IND-CCA 安全性が, これまでそのように明示的に定義されたことはなかったが, 実際には“内部者安全性”であることを説明する.

まず, Signcryption の秘匿性の多人数モデルにおける内部者安全性に関して, Matsuda ら [30] は, 2 種類の定義を用いている. 一つは, 攻撃者がチャレンジサインクリプテキスト生成に用いる送信者の秘密鍵を任意に選択可能な定義であり, 当該論文ではこれを, “indistinguishability against insider chosen ciphertext attacks in the dynamic multi-user model(dM-IND-CCA)” と呼んでいる. もう一つは, 攻撃者はチャレンジサインクリプテキスト生成に用いる送信者の秘密鍵を任意に選択可能ではないが, 挑戦者にそれを教えてもらえる定義であり, 当該論文ではこれを, “indistinguishability against insider chosen ciphertext attacks in the fixed multi-user model(fM-IND-CCA)” と呼んでいる. ちなみに, 本研究で CP-ABSC の一般的構成法を考える上で構成法を参考にしている Chiba らの論文 [31] では, 前者の定義を用いている. ここで, 前者が後者よりも真に強い安全性であることは, 定義より自明である. そして, ともに内部者安全性である両者に共通しているのは, チャレンジサインクリプテキストの送信者がチャレンジサインクリプテキストを生成するのに必要な情報と同等の情報 (具体的には, 送信者の秘密鍵, 受信者の公開鍵) を攻撃者は知ることができる, という点である.

次に, CP-ABSC の AP-IND-CCA 安全性が内部者安全性であるかどうかについて考える. 2.8.1 項の AP-IND-CCA 安全性の定義において, 注目すべき点は, 攻撃者が送信者用鍵生成オラクルを利用する上で, 如何なる制限も課されていない点である. よって, 攻撃者は, チャレンジフェーズで如何なるターゲット署名者アクセス構造  $A_s^*$  を選ぼうとも,  $S_s \in A_s^*$  を満たす任意の属性集合  $S_s$  について,  $S_s$  を送信者用鍵生成オラクルへクエリして対応する秘密鍵を入手することが可能である. 従って, 攻撃者は, チャレンジサインクリプテキスト生成に必要な情報と同等の情報を, 入手可能である. ゆえに, CP-ABSC の AP-IND-CCA 安全性は, 実際には内部者安全性であると言える.

次節の SP-IND-CCA 安全性も AP-IND-CCA 安全性と同様の理由で内部者安全性であると言える.

## 2.8.2 SP-IND-CCA 安全性

選択的述語モデルにおける適応的選択暗号文 (サインクリプテキスト) 攻撃に対する暗号文 (サインクリプテキスト) 識別不可能性 (Ciphertext(Signcryptext)-Indistinguishability under Adaptively Chosen Ciphertext(Signcryptext) Attacks in the Selective Predicate Model, AP-IND-CCA) は, CP-ABSC 方式が満たすべき秘匿性に関する安全性の中で, AP-IND-CCA よりも弱いことが自明な安全性である. CP-ABSC 方式  $\Pi_{CS}$  に関する SP-IND-CCA 安全性は, 攻撃者  $\mathcal{A}$  と挑戦者  $\mathcal{CH}$  との間で行われる以下の SP-IND-CCA 安全性ゲームによって定義する.

**Init Phase:**  $\mathcal{CH}$  はターゲット復号者アクセス構造  $A_d^*$  を  $\mathcal{CH}$  へ送る.

**Setup Phase:**  $\mathcal{CH}$  が  $CS.Setup(1^k, \mathcal{U}_s, \mathcal{U}_r) \rightarrow (PK, MK)$  を実行し, システム公開鍵 PK を  $\mathcal{A}$  に渡す.

**Query Phase 1:**  $\mathcal{A}$  は以下の各オラクルに対してクエリを任意回数発行することができる

**送信者用鍵生成:**  $\mathcal{A}$  は属性集合  $S_s \in (2^{U_s} - \{\phi\})$  をクエリする.  $\mathcal{CH}$  は  $\text{CS.KeyGen}_S(\text{PK}, \text{MK}, S_s) \rightarrow \text{SK}_s$  を実行し  $\text{SK}_s$  を  $\mathcal{CH}$  へ送る.

**受信者用鍵生成:**  $\mathcal{A}$  は属性集合  $S_r \in (2^{U_r} - \{\phi\})$  をクエリする. 但し,  $S_r \in \mathbb{A}_d^*$  を満たす  $S_r$  のクエリは禁止.  $\mathcal{CH}$  は  $\text{KeyGen}_R(\text{PK}, \text{MK}, S_r) \rightarrow \text{SK}_r$  を実行し  $\text{SK}_r$  を  $\mathcal{CH}$  へ送る.

**サインクリプションオラクル:**  $\mathcal{A}$  は平文  $m \in \mathcal{M}$ , 署名者アクセス構造  $\mathbb{A}_s$ , 復号者アクセス構造  $\mathbb{A}_d$ ,  $S_s \in \mathbb{A}_s$  を満たす属性集合  $S_s \in (2^{U_s} - \{\phi\})$  をクエリする.  $\mathcal{CH}$  は  $\text{CS.KeyGen}_S(\text{PK}, \text{MK}, S_s) \rightarrow \text{SK}_s$ ,  $\text{CS.SC}(\text{PK}, m, \text{SK}_s, \mathbb{A}_s, \mathbb{A}_d) \rightarrow C$  を実行し,  $C$  を  $\mathcal{A}$  へ送る.

**アンサインクリプションオラクル:**  $\mathcal{A}$  はサインクリプテキスト  $C \in \mathcal{C}$ , 署名者アクセス構造  $\mathbb{A}_s$ , 属性集合  $S_r \in (2^{U_r} - \{\phi\})$  をクエリする.  $\mathcal{CH}$  は  $\text{CS.KeyGen}_R(\text{PK}, \text{MK}, S_r) \rightarrow \text{SK}_r$ ,  $\text{CS.USC}(\text{PK}, C, \text{SK}_r, \mathbb{A}_s) \rightarrow m/\perp$  を実行し最終実行結果を  $\mathcal{A}$  へ送る.

**Challenge Phase:**  $\mathcal{A}$  は署名者アクセス構造  $\mathbb{A}_s^*$ , 属性集合  $S_s^* \in \mathbb{A}_s^*$ , 及びサイズの等しい二つの平文  $m_0 \in \mathcal{M}, m_1 \in \mathcal{M}$  を選び  $\mathcal{CH}$  に送る.  $\mathcal{CH}$  は  $\text{SK}_s^* \leftarrow \text{CS.KeyGen}_S(\text{PK}, \text{MK}, S_s^*), b \xleftarrow{\text{U}} \{0, 1\}, C^* \leftarrow \text{CS.SC}(\text{PK}, m_b^*, \text{SK}_s^*, \mathbb{A}_s^*, \mathbb{A}_d^*)$  を実行し, チャレンジサインクリプテキスト  $C^*$  を  $\mathcal{A}$  に送る.

**Query Phase 2:**  $\mathcal{A}$  は以下の各オラクルに対してクエリを任意回数発行することができる.

**送信者用鍵生成:** Query Phase 1 と同様.

**受信者用鍵生成:** Query Phase 1 と同様. 但し,  $S_r \in \mathbb{A}_d^*$  を満たす  $S_r$  のクエリは禁止.

**サインクリプションオラクル:** Query Phase 1 と同様.

**アンサインクリプションオラクル:** Query Phase 1 と同様. 但し,  $(C, \mathbb{A}_s, S_r) = (C^*, \mathbb{A}_s^*, S_r (s.t. S_r \in \mathbb{A}_d^*))$  のクエリは禁止.

**Guess Phase:**  $\mathcal{A}$  はチャレンジビット  $b$  に対する推測として  $b' \in \{0, 1\}$  を出力.

CP-ABSC 方式  $\Pi_{\text{CS}}$  に関する SP-IND-CCA 安全性ゲームにおいて攻撃者  $\mathcal{A}$  の優位性を次式で定義する.

$$\text{Adv}_{\Pi_{\text{CS}}, \mathcal{A}}^{\text{SP-IND-CCA}} = |\Pr[b' = b] - \frac{1}{2}| \quad (2.34)$$

**定義 2.28.** 全ての PPTA 攻撃者  $\mathcal{A}$  に対して, CP-ABSC 方式  $\Pi_{\text{CS}}$  に関する SP-IND-CCA 安全性ゲームにおける  $\mathcal{A}$  の優位性  $\text{Adv}_{\Pi_{\text{CS}}, \mathcal{A}}^{\text{SP-IND-CCA}}$  が, セキュリティパラメータ  $k$  に関

して無視できるほど小さい値であるならば,  $CP\text{-}ABSC$  方式  $\Pi_{CS}$  は  $SP\text{-}IND\text{-}CCA$  安全である.

### 2.8.3 AP-sEUF-CMA 安全性

適応的述語モデルにおける適応的選択文書攻撃に対するサインクリプテキスト強偽造不可能性 (Strongly Existentially Unforgeability of Signcryptext under Adaptively Chosen Message Attacks in the Adaptive Predicate model, AP-sEUF-CMA) は,  $CP\text{-}ABSC$  方式が満たすべき完全性に関する安全性の中で, 最も強い安全性である.  $CP\text{-}ABSC$  方式  $\Pi_{CS}$  に関する AP-sEUF-CMA 安全性は, 攻撃者  $\mathcal{A}$  と挑戦者  $\mathcal{CH}$  との間で行われる以下の AP-sEUF-CMA 安全性ゲームによって定義する.

**Setup Phase:**  $\mathcal{CH}$  が  $CS.Setup(1^k, \mathcal{U}_s, \mathcal{U}_r) \rightarrow (PK, MK)$  アルゴリズムを実行し, システム公開鍵  $PK$  を  $\mathcal{A}$  に渡す.

**Query Phase:**  $\mathcal{A}$  は以下の各オラクルに対してクエリを任意回数発行することができる.

**送信者用鍵生成:**  $\mathcal{A}$  は属性集合  $S_s \in (2^{\mathcal{U}_s} - \{\phi\})$  をクエリする.  $\mathcal{CH}$  は  $CS.KeyGen_s(PK, MK, S_s) \rightarrow SK_s$  を実行し  $SK_s$  を  $\mathcal{A}$  へ送る. 但し,  $S_s \in \mathcal{A}_s^*$  を満たす  $S_s$  をクエリすることは禁止とする.

**受信者用鍵生成:**  $\mathcal{A}$  は属性集合  $S_r \in (2^{\mathcal{U}_r} - \{\phi\})$  をクエリする.  $\mathcal{CH}$  は  $CS.KeyGen_r(PK, MK, S_r) \rightarrow SK_r$  を実行し  $SK_r$  を  $\mathcal{A}$  へ送る.

**サインクリプション:**  $\mathcal{A}$  は平文  $m \in \mathcal{M}$ , 署名者アクセス構造  $\mathbb{A}_s$ , 復号者アクセス構造  $\mathbb{A}_d$ ,  $S_s \in \mathbb{A}_s$  を満たす属性集合  $S_s \in (2^{\mathcal{U}_s} - \{\phi\})$  を選び,  $\mathcal{CH}$  へ送る.  $\mathcal{CH}$  は  $CS.KeyGen_s(PK, MK, S_s) \rightarrow SK_s$ ,  $CS.SC(PK, m, SK_s, \mathbb{A}_s, \mathbb{A}_d) \rightarrow C$  を実行し,  $C$  を  $\mathcal{A}$  へ送る.  $\mathcal{CH}$  は  $(m, C, \mathbb{A}_s, \mathbb{A}_d)$  を  $\mathcal{L}_{CS.SC}$  へ追加する.

**アンサインクリプション:**  $\mathcal{A}$  はサインクリプテキスト  $C \in \mathcal{C}$ , 署名者アクセス構造  $\mathbb{A}_s$ , 属性集合  $S_r$  を  $\mathcal{CH}$  へ送る.  $\mathcal{CH}$  は  $CS.KeyGen_r(PK, MK, S_r) \rightarrow SK_r$ ,  $CS.SC(PK, C, SK_r, \mathbb{A}_s) \rightarrow m/\perp$  を実行し最後の実行結果を  $\mathcal{A}$  へ送る.

**Forgery:**  $\mathcal{A}$  が  $(C^*, \mathbb{A}_s^*, \mathbb{A}_d^*)$  を出力.

$CP\text{-}ABSC$  方式  $\Pi_{CS}$  に関する AP-sEUF-CMA 安全性ゲームにおいて攻撃者  $\mathcal{A}$  の優位性を次式で定義する. 次式中,  $Disclose_{CS}$  は,  $\Pi_{CS}$  の復号者アクセス構造開示性アルゴリズムであるとする.

$$\begin{aligned}
& \text{Adv}_{\Pi_{\text{CS}}, \mathcal{A}}^{\text{AP-sEUF-CMA}} \\
= & \Pr[[\text{Disclose}_{\text{CS}}(C^*) = \mathbb{A}_d^*] \\
& \wedge [\forall S_r^{(i)} \in \mathbb{A}_d^*, \text{CS.KeyGen}_R(\text{PK}, \text{MK}, S_r^{(i)}) \rightarrow \text{SK}_r^{(i)}, \\
& \text{CS.USC}(\text{PK}, C^*, \text{SK}_r^{(i)}, \mathbb{A}_s^*) \rightarrow m^{(i)} \in \mathcal{M}] \\
& \wedge [m^{(1)} = \dots = m^{(|\mathbb{A}_d^*|)} =: m^*] \wedge \underline{[(m^*, C^*, \mathbb{A}_s^*, \mathbb{A}_d^*) \notin \mathcal{L}_{\text{CS.sc}}]}] \quad (2.35)
\end{aligned}$$

**定義 2.29.** 全ての  $PPTA$  攻撃者  $\mathcal{A}$  に対して,  $CP\text{-}ABSC$  方式  $\Pi_{\text{CS}}$  に関する  $AP\text{-}sEUF\text{-}CMA$  安全性ゲームにおける  $\mathcal{A}$  の優位性  $\text{Adv}_{\Pi_{\text{CS}}, \mathcal{A}}^{\text{AP-sEUF-CMA}}$  が, セキュリティパラメータ  $k$  に関して無視できるほど小さい値であるならば,  $CP\text{-}ABSC$  方式  $\Pi_{\text{CS}}$  は  $AP\text{-}sEUF\text{-}CMA$  安全である.

定義 2.29 は, “強偽造不可能性” である. それに対して, “弱偽造不可能性”, 正式には, 適応的述語モデルにおける適応的選択文書攻撃に対するサインクリプテキスト強偽造不可能性 (Strongly Existentially Unforgeability of Signcryptext under Adaptively Chosen Message Attacks in the Adaptive Predicate model,  $AP\text{-}wEUF\text{-}CMA$ ) は, 先の  $AP\text{-}sEUF\text{-}CMA$  安全性ゲームと同一の  $AP\text{-}wEUF\text{-}CMA$  安全性ゲームにおいて, 攻撃者の優位性を次式で定義する. ここで, 式 (2.35) と式 (2.36) の違いは, 下線部分である.

$$\begin{aligned}
& \text{Adv}_{\Pi_{\text{CS}}, \mathcal{A}}^{\text{AP-wEUF-CMA}} \\
= & \Pr[[\text{Disclose}_{\text{CS}}(C^*) = \mathbb{A}_d^*] \\
& \wedge [\forall S_r^{(i)} \in \mathbb{A}_d^*, \text{CS.KeyGen}_R(\text{PK}, \text{MK}, S_r^{(i)}) \rightarrow \text{SK}_r^{(i)}, \\
& \text{CS.USC}(\text{PK}, C^*, \text{SK}_r^{(i)}, \mathbb{A}_s^*) \rightarrow m^{(i)} \in \mathcal{M}] \\
& \wedge [m^{(1)} = \dots = m^{(|\mathbb{A}_d^*|)} =: m^*] \wedge \underline{[(m^*, \mathbb{A}_s^*, \mathbb{A}_d^*) \notin \mathcal{L}_{\text{CS.sc}}]}] \quad (2.36)
\end{aligned}$$

**定義 2.30.** 全ての  $PPTA$  攻撃者  $\mathcal{A}$  に対して,  $CP\text{-}ABSC$  方式  $\Pi_{\text{CS}}$  に関する  $AP\text{-}wEUF\text{-}CMA$  安全性ゲームにおける  $\mathcal{A}$  の優位性  $\text{Adv}_{\Pi_{\text{CS}}, \mathcal{A}}^{\text{AP-wEUF-CMA}}$  が, セキュリティパラメータ  $k$  に関して無視できるほど小さい値であるならば,  $CP\text{-}ABSC$  方式  $\Pi_{\text{CS}}$  は  $AP\text{-}wEUF\text{-}CMA$  安全である.

**AP-sEUF-CMA 安全性は, “内部者安全性” である.**

本パラグラフでは,  $CP\text{-}ABSC$  の  $AP\text{-}sEUF\text{-}CMA$  安全性が, これまでそのような明示的な定義が行われたことはなかったが, 実際には “内部者安全性” であることを説明する.

まず, **Signcryption** の偽造不可能性の多人数モデルにおける内部者安全性に関して, Matsuda ら [30] は, 2 種類の定義を用いている. 一つは, 偽造サインクリプテキストをアンサインクリプションする際に用いる受信者の秘密鍵を攻撃者が任意に選択可能な定義であり, 当該論文ではこれを, “strong unforgeability against insider chosen message attacks in the dynamic multi-user model( $DM\text{-}sEUF\text{-}CMA$ )” と呼んでいる. もう一つは,

偽造サインクリプトテキストをアンサインクリプションする際に用いる受信者の秘密鍵を攻撃者が選択することはできないが、攻撃者はそれを挑戦者に教えてもらえる定義であり、当該論文ではこれを、“strong unforgeability against insider chosen message attacks in the fixed multi-user model(fM-sEUF-CMA)”と呼んでいる。ちなみに、本研究でCP-ABSCの一般的構成法を考える上で構成法を参考にしているChibaらの論文[31]では、前者の定義を用いている。ここで、前者が後者よりも真に強い安全性であることは、定義より自明である。そして、ともに内部者安全性である両者に共通しているのは、チャレンジサインクリプトテキストの受信者がチャレンジサインクリプトテキストをアンサインクリプションするのに必要な情報と同等の情報（具体的には、受信者の秘密鍵、送信者の公開鍵）を攻撃者は知ることができる、という点である。

次に、CP-ABASCのAP-sEUF-CMA安全性が内部者安全性であるかどうかについて考える。2.8.3項のAP-sEUF-CMA安全性の定義において、注目すべき点は、攻撃者が受信者用鍵生成オラクルを利用する上で、如何なる制限も課されていない点である。よって、攻撃者は、Forgeryフェーズで如何なるターゲット復号者アクセス構造 $A_d^*$ を選ぼうとも、 $S_r \in A_d^*$ を満たす任意の属性集合 $S_r$ について、 $S_r$ を受信者用鍵生成オラクルへクエリして対応する秘密鍵を入手することが可能である。従って、攻撃者は、チャレンジサインクリプトテキストをアンサインクリプションするのに必要な情報と同等の情報を、入手可能である。ゆえに、CP-ABSCのAP-sEUF-CMA安全性は、実際には内部者安全性であると言える。

本節のAP-wEUF-CMA、次節のSP-sEUF-CMA、SP-wEUF-CMAも、AP-sEUF-CMA安全性が内部者安全性であるのと同様の理由で内部者安全性であると言える。

## 2.8.4 SP-sEUF-CMA 安全性

選択的述語モデルにおける適応的選択文書攻撃に対するサインクリプトテキスト強偽造不可能性（Strongly Existentially Unforgeability of Signcryptext under Adaptively Chosen Message Attacks in the Selective Predicate model, SP-sEUF-CMA）は、CP-ABSC方式が満たすべき完全性に関する安全性の中で、SP-sEUF-CMA(2.8.3項)よりも弱いことが自明な安全性である。CP-ABSC方式 $\Pi_{CS}$ に関するSP-sEUF-CMA安全性は、攻撃者 $\mathcal{A}$ と挑戦者 $\mathcal{CH}$ との間で行われる以下のSP-sEUF-CMA安全性ゲームによって定義する。

**Init Phase:**  $\mathcal{CH}$ はターゲット署名者アクセス構造 $A_s^*$ を $\mathcal{CH}$ へ送る。

**Setup Phase:**  $\mathcal{CH}$ が $CS.Setup(1^k, \mathcal{U}_s, \mathcal{U}_r) \rightarrow (PK, MK)$ アルゴリズムを実行し、システム公開鍵 $PK$ を $\mathcal{A}$ に渡す。

**Query Phase:**  $\mathcal{A}$ は以下の各オラクルに対してクエリを任意回数発行することができる。

**送信者用鍵生成:**  $\mathcal{A}$ は属性集合 $S_s \in (2^{\mathcal{U}_s} - \{\phi\})$ をクエリする。 $\mathcal{CH}$ は $CS.KeyGen_s(PK, MK, S_s) \rightarrow SK_s$ を実行し $SK_s$ を $\mathcal{A}$ へ送る。但し、

$S_s \in \mathbb{A}_s^*$  を満たす  $S_s$  をクエリすることは禁止とする。

**受信者用鍵生成:**  $\mathcal{A}$  は属性集合  $S_r \in (2^{\mathcal{U}_r} - \{\phi\})$  をクエリする。  $\mathcal{CH}$  は  $\text{CS.KeyGen}_R(\text{PK}, \text{MK}, S_r) \rightarrow \text{SK}_r$  を実行し  $\text{SK}_r$  を  $\mathcal{A}$  へ送る。

**サインクリプション:**  $\mathcal{A}$  は平文  $m \in \mathcal{M}$ , 署名者アクセス構造  $\mathbb{A}_s$ , 復号者アクセス構造  $\mathbb{A}_d$ ,  $S_s \in \mathbb{A}_s$  を満たす属性集合  $S_s \in (2^{\mathcal{U}_s} - \{\phi\})$  を選び,  $\mathcal{CH}$  へ送る。  $\mathcal{CH}$  は  $\text{CS.KeyGen}_S(\text{PK}, \text{MK}, S_s) \rightarrow \text{SK}_s$ ,  $\text{CS.SC}(\text{PK}, m, \text{SK}_s, \mathbb{A}_s, \mathbb{A}_d) \rightarrow C$  を実行し,  $C$  を  $\mathcal{A}$  へ送る。  $\mathcal{CH}$  は  $(m, C, \mathbb{A}_s, \mathbb{A}_d)$  を  $\mathcal{L}_{\text{CS.SC}}$  へ追加する。

**アンサインクリプション:**  $\mathcal{A}$  はサインクリプテキスト  $C \in \mathcal{C}$ , 署名者アクセス構造  $\mathbb{A}_s$ , 属性集合  $S_r$  を  $\mathcal{CH}$  へ送る。  $\mathcal{CH}$  は  $\text{CS.KeyGen}_R(\text{PK}, \text{MK}, S_r) \rightarrow \text{SK}_r$ ,  $\text{CS.SC}(\text{PK}, C, \text{SK}_r, \mathbb{A}_s) \rightarrow m/\perp$  を実行し最後の実行結果を  $\mathcal{A}$  へ送る。

**Forgery:**  $\mathcal{A}$  が  $(C^*, \mathbb{A}_d^*)$  を出力。

CP-ABSC 方式  $\Pi_{\text{CS}}$  に関する SP-sEUF-CMA 安全性ゲームにおいて攻撃者  $\mathcal{A}$  の優位性を次式で定義する。なお, 次式中,  $\text{Disclose}_{\text{CS}}$  は CP-ABSC 方式  $\Pi_{\text{CS}}$  の復号者アクセス構造開示性である。

$$\begin{aligned}
& \text{Adv}_{\Pi_{\text{CS}}, \mathcal{A}}^{\text{SP-sEUF-CMA}} \\
&= \Pr[[\text{Disclose}_{\text{CS}}(C^*) = \mathbb{A}_d^*] \\
&\quad \wedge [\forall S_r^{(i)} \in \mathbb{A}_d^*, \text{CS.KeyGen}_R(\text{PK}, \text{MK}, S_r^{(i)}) \rightarrow \text{SK}_r^{(i)}, \\
&\quad \quad \text{CS.USC}(\text{PK}, C^*, \text{SK}_r^{(i)}, \mathbb{A}_s^*) \rightarrow m^{(i)} \in \mathcal{M}] \\
&\quad \wedge [m^{(1)} = \dots = m^{(|\mathbb{A}_d^*|)} =: m^*] \wedge [(m^*, C^*, \mathbb{A}_s^*, \mathbb{A}_d^*) \notin \mathcal{L}_{\text{CS.SC}}]] \quad (2.37)
\end{aligned}$$

**定義 2.31.** 全ての PPTA 攻撃者  $\mathcal{A}$  に対して, CP-ABSC 方式  $\Pi_{\text{CS}}$  に関する SP-sEUF-CMA 安全性ゲームにおける  $\mathcal{A}$  の優位性  $\text{Adv}_{\Pi_{\text{CS}}, \mathcal{A}}^{\text{SP-sEUF-CMA}}$  が, セキュリティパラメータ  $k$  に関して無視できるほど小さい値であるならば, CP-ABSC 方式  $\Pi_{\text{CS}}$  は SP-sEUF-CMA 安全である。

定義 2.31 は, “強偽造不可能性” である。それに対して, “弱偽造不可能性”, 正式には, 適応的述語モデルにおける適応的選択文書攻撃に対するサインクリプテキスト弱偽造不可能性 (Weakly Existentially Unforgeability of Signcryptext under Adaptively Chosen Message Attacks in the Adaptive Predicate model, AP-wEUF-CMA) は, 先の SP-sEUF-CMA 安全性ゲームと同一の SP-wEUF-CMA 安全性ゲームにおいて, 攻撃者の優位性を次式で定義する。ここで, 式 (2.37) と式 (2.38) の違いは, 下線部分である。

$$\begin{aligned}
& \text{Adv}_{\Pi_{\text{CS}}, \mathcal{A}}^{\text{SP-wEUF-CMA}} \\
= & \Pr[[\text{Disclose}_{\text{CS}}(C^*) = \mathbb{A}_d^*] \\
& \wedge [\forall S_r^{(i)} \in \mathbb{A}_d^*, \text{CS.KeyGen}_R(\text{PK}, \text{MK}, S_r^{(i)}) \rightarrow \text{SK}_r^{(i)}, \\
& \quad \text{CS.USC}(\text{PK}, C^*, \text{SK}_r^{(i)}, \mathbb{A}_s^*) \rightarrow m^{(i)} \in \mathcal{M}] \\
& \wedge [m^{(1)} = \dots = m^{(\mathbb{A}_d^*)} =: m^*] \wedge [(m^*, \mathbb{A}_s^*, \mathbb{A}_d^*) \notin \mathcal{L}_{\text{CS.sc}}]] \quad (2.38)
\end{aligned}$$

**定義 2.32.** 全ての  $PPTA$  攻撃者  $\mathcal{A}$  に対して,  $CP\text{-}ABSC$  方式  $\Pi_{\text{CS}}$  に関する  $SP\text{-}wEUF\text{-}CMA$  安全性ゲームにおける  $\mathcal{A}$  の優位性  $\text{Adv}_{\Pi_{\text{CS}}, \mathcal{A}}^{\text{SP-wEUF-CMA}}$  が, セキュリティパラメータ  $k$  に関して無視できるほど小さい値であるならば,  $CP\text{-}ABSC$  方式  $\Pi_{\text{CS}}$  は  $SP\text{-}wEUF\text{-}CMA$  安全である.

## 2.8.5 完全匿名性

$CP\text{-}ABSC$  方式の完全匿名性 (Perfect Privacy, あるいは Signer Privacy) は, 直感的な説明としては, サインクリプトテキストからサインクリプタの属性集合が漏れないことを保障する安全性である. 具体的には, 次のように定義される.

**定義 2.33.**  $CP\text{-}ABSC$  方式が完全匿名性を満たすとは, 当該方式が以下の条件を満たす場合を言う. :

全ての  $(\text{PK}, \text{MK}) \leftarrow \text{CS.Setup}(1^k, \mathcal{U}_s, \mathcal{U}_r)$ , 全ての  $S_s \in (2^u - \{\phi\})$ , 全ての  $S'_s \in (2^u - \{\phi\})$ , 全ての  $\text{SK}_s \leftarrow \text{CS.KeyGen}_s(\text{PK}, \text{MK}, S_s)$ , 全ての  $\text{SK}'_s \leftarrow \text{CS.KeyGen}_s(\text{PK}, \text{MK}, S'_s)$ , 全ての  $m \in \mathcal{M}$ , 全ての  $\mathbb{A}_s (s.t. S_s \in \mathbb{A}_s \wedge S'_s \in \mathbb{A}_s)$ , 全ての  $\mathbb{A}_d$  に対して,  $\text{CS.SC}(\text{PK}, m, \text{SK}_s, \mathbb{A}_s, \mathbb{A}_d)$  の確率分布と  $\text{CS.SC}(\text{PK}, m, \text{SK}'_s, \mathbb{A}_s, \mathbb{A}_d)$  の確率分布が同一である.

## 2.8.6 復号者アクセス構造開示性

$CP\text{-}ABSC$  方式の“復号者アクセス構造開示性”は著者が独自に命名し, かつ定義した性質である. 直感的には, サインクリプトテキストからサインクリプトテキスト生成に使用された (あるいは, サインクリプトテキストと真に関連付けられた) 復号者アクセス構造を完璧に復元できることを意味する. 具体的には, 次の様に定義される.

**定義 2.34.**  $CP\text{-}ABSC$  方式が復号者アクセス構造開示性を満たすとは,  $PPTA \text{ Disclose}_{\text{CS}}$  が存在し, 全ての  $k$ , 全ての  $\mathcal{U}_s$ , 全ての  $\mathcal{U}_r$ , 全ての  $(\text{PK}, \text{MK}) \leftarrow \text{CS.Setup}(1^k, \mathcal{U}_s, \mathcal{U}_r)$ , 全ての  $m$ , 全ての  $S_s \in (2^u - \{\phi\})$ , 全ての  $\text{SK}_s \leftarrow \text{CS.KeyGen}_s(\text{PK}, \text{MK}, S_s)$ , 全ての  $\mathbb{A}_s (s.t. S_s \in \mathbb{A}_s)$ , 全ての  $\mathbb{A}_d$ , 全ての  $C \leftarrow \text{CS.SC}(\text{PK}, m, \text{SK}_s, \mathbb{A}_s, \mathbb{A}_d)$  に対して, 以下の条件式が成立する場合を言う.

$$\Pr[\mathbb{A}'_d = \mathbb{A}_d | \mathbb{A}'_d \leftarrow \text{Disclose}_{\text{CS}}(\text{PK}, C)] = 1 \quad (2.39)$$

$CP\text{-}ABSC$  方式の“復号者アクセス構造開示性”は特殊な性質ではなく極めて自然な性質である. 事実として著者が知る限り, 既存の  $CP\text{-}ABSC$  方式の多数 [1][2][4][5][9][32] に関して, 容易に定義 2.34 の条件を満たすような  $PPTA \text{ Disclose}_{\text{CS}}$  を構成できる.

## 2.9 鍵ポリシー型属性ベース Signcryption

鍵ポリシー型属性ベース Signcryption (Key-Policy Attribute-Based Signcryption, 以降略称として“KP-ABSC”と表記することもある) は, KP-ABE と KP-ABS の両機能を実現する暗号技術である. KP-ABSC は以下の 5 つのアルゴリズムから構成される.

$\text{KSC.Setup}(1^k, \mathcal{U}_s, \mathcal{U}_r) \rightarrow (\text{PK}, \text{MK})$

セットアップアルゴリズム  $\text{KSC.Setup}$  は, セキュリティパラメータ  $k$ , 送信者の属性の全体集合  $\mathcal{U}_s$ , 受信者の属性の全体集合  $\mathcal{U}_r$  を入力とし, システム公開鍵  $\text{PK}$  とマスター秘密鍵  $\text{MK}$  を出力する.

$\text{KSC.KeyGen}_S(\text{PK}, \text{MK}, S_s) \rightarrow \text{SK}_s$

送信者用秘密鍵生成アルゴリズム  $\text{KSC.KeyGen}_S$  は, システム公開鍵  $\text{PK}$ , マスター秘密鍵  $\text{MK}$ , 送信者のポリシー  $\mathbb{A}_s$  を入力とし, 送信者用秘密鍵  $\text{SK}_s$  を出力する.

$\text{KSC.KeyGen}_R(\text{PK}, \text{MK}, S_r) \rightarrow \text{SK}_r$

受信者用秘密鍵生成アルゴリズム  $\text{KSC.KeyGen}_R$  は, システム公開鍵  $\text{PK}$ , マスター秘密鍵  $\text{MK}$ , 受信者のポリシー  $\mathbb{A}_r$  を入力とし, 受信者用秘密鍵  $\text{SK}_r$  を出力する.

$\text{KSC.SC}(\text{PK}, m, \text{SK}_s, S_s, S_d) \rightarrow C$

サインクリプションアルゴリズム  $\text{SC}$  は, システム公開鍵  $\text{PK}$ , 平文  $m \in \mathcal{M}$ , 送信者用秘密鍵  $\text{SK}_s$ , 署名者属性集合  $S_s$ , 復号者属性集合  $S_d$  を入力とし, サインクリプトテキスト  $C \in \mathcal{C}$  を出力する.

$\text{USC}(\text{PK}, C, \text{SK}_r, S_s) \rightarrow m / \perp$

アンサインクリプションアルゴリズム  $\text{KSC.USC}$  は, システム公開鍵  $\text{PK}$ , サインクリプトテキスト  $C \in \mathcal{C}$ , 署名者属性集合  $\mathbb{A}_s$ , 受信者用秘密鍵  $\text{SK}_r$  を入力とし, 平文  $m \in \mathcal{M}$  または  $\perp$  を出力する.

**正当性 (Correctness)** KP-ABSC 方式は正当であることが求められる. KP-ABSC 方式が正当であるとは, 全ての  $k$ , 全ての  $\mathcal{U}_s$ , 全ての  $\mathcal{U}_r$ , 全ての  $(\text{PK}, \text{MK}) \leftarrow \text{KSC.Setup}(1^k, \mathcal{U}_s, \mathcal{U}_r)$ , 全ての  $\mathbb{A}_s$ , 全ての  $\text{SK}_s \leftarrow \text{KSC.KeyGen}_S(\text{PK}, \text{MK}, \mathbb{A}_s)$ ,  $S_s \in \mathbb{A}_s$  を満たす全ての  $S_s$ , 全ての  $\mathbb{A}_r$ , 全ての  $\text{SK}_r \leftarrow \text{KSC.KeyGen}_R(\text{PK}, \text{MK}, \mathbb{A}_r)$ ,  $S_d \in \mathbb{A}_r$  を満たす全ての  $S_d$ , 全ての  $m \in \mathcal{M}$ , 全ての  $C \leftarrow \text{KSC.SC}(\text{PK}, m, \text{SK}_s, S_s, S_d)$  に対して, 次の等式が成立する場合を言う.

$$\Pr[\text{KSC.USC}(\text{PK}, C, \text{SK}_r, S_s) = m] = 1 \quad (2.40)$$

### 2.9.1 AA-IND-CCA 安全性

適応的属性モデルにおける適応的選択暗号文(サインクリプトテキスト)攻撃に対する暗号文(サインクリプトテキスト)識別不可能性 (Ciphertext(Signcryptext)-Indistinguishability

under Adaptively Chosen Ciphertext(Signcryptext) Attacks in the Adaptive Attribute Model, AA-IND-CCA) は, KP-ABSC 方式が満たすべき秘匿性に関する安全性の中で, 最も強いと考えられている安全性である. KP-ABSC 方式  $\Pi_{\text{KSC}}$  に関する AA-IND-CCA 安全性は, 攻撃者  $\mathcal{A}$  と挑戦者  $\mathcal{CH}$  との間で行われる以下の AA-IND-CCA 安全性ゲームによって定義する.

**Setup Phase:**  $\mathcal{CH}$  が  $\text{KSC.Setup}(1^k, \mathcal{U}_s, \mathcal{U}_r) \rightarrow (\text{PK}, \text{MK})$  を実行し, システム公開鍵  $\text{PK}$  を  $\mathcal{A}$  に渡す.

**Query Phase 1:**  $\mathcal{A}$  は以下の各オラクルに対してクエリを任意回数発行することができる

**送信者用鍵生成:**  $\mathcal{A}$  は送信者のポリシー  $A_s$  をクエリする.  $\mathcal{CH}$  は  $\text{KSC.KeyGen}_s(\text{PK}, \text{MK}, A_s) \rightarrow \text{SK}_s$  を実行し  $\text{SK}_s$  を  $\mathcal{CH}$  へ送る.

**受信者用鍵生成:**  $\mathcal{A}$  は受信者のポリシー  $A_r$  をクエリする. 但し, 後に Challenge Phase で選択する  $S_d^*$  に関して,  $S_d^* \in A_r$  を満たす  $A_r$  のクエリは禁止.  $\mathcal{CH}$  は  $\text{KeyGen}_R(\text{PK}, \text{MK}, A_r) \rightarrow \text{SK}_r$  を実行し  $\text{SK}_r$  を  $\mathcal{CH}$  へ送る.

**サインクリプションオラクル:**  $\mathcal{A}$  は平文  $m \in \mathcal{M}$ , 署名者属性集合  $S_s$ , 復号者属性集合  $S_d$ ,  $S_s \in A_s$  を満たすポリシー  $A_s$  をクエリする.  $\mathcal{CH}$  は  $\text{KSC.KeyGen}_s(\text{PK}, \text{MK}, A_s) \rightarrow \text{SK}_s$ ,  $\text{KSC.SC}(\text{PK}, m, \text{SK}_s, S_s, S_d) \rightarrow C$  を実行し,  $C$  を  $\mathcal{A}$  へ送る.

**アンサインクリプションオラクル:**  $\mathcal{A}$  はサインクリプトテキスト  $C \in \mathcal{C}$ , 署名者属性集合  $S_s$ , ポリシー  $A_r$  をクエリする.  $\mathcal{CH}$  は  $\text{KSC.KeyGen}_R(\text{PK}, \text{MK}, A_r) \rightarrow \text{SK}_r$ ,  $\text{KSC.USC}(\text{PK}, C, \text{SK}_r, S_s) \rightarrow m/\perp$  を実行し最終実行結果を  $\mathcal{A}$  へ送る.

**Challenge Phase:**  $\mathcal{A}$  は復号者属性集合  $S_d^*$ , 署名者属性集合  $S_s^*$ ,  $S_s^* \in A_s$  を満たすポリシー  $A_s$ , 及びサイズの等しい二つの平文  $m_0 \in \mathcal{M}, m_1 \in \mathcal{M}$  を選び  $\mathcal{CH}$  に送る.  $\mathcal{CH}$  は  $\text{SK}_s^* \leftarrow \text{KSC.KeyGen}_s(\text{PK}, \text{MK}, A_s), b \xleftarrow{\text{U}} \{0, 1\}, C^* \leftarrow \text{KSC.SC}(\text{PK}, m_b^*, \text{SK}_s^*, S_s^*, S_d^*)$  を実行し, チャレンジサインクリプトテキスト  $C^*$  を  $\mathcal{A}$  に送る.

**Query Phase 2:**  $\mathcal{A}$  は以下の各オラクルに対してクエリを任意回数発行することができる.

**送信者用鍵生成:** Query Phase 1 と同様.

**受信者用鍵生成:** Query Phase 1 と同様. 但し,  $S_d^* \in A_r$  を満たす  $A_r$  のクエリは禁止.

**サインクリプションオラクル:** Query Phase 1 と同様.

**アンサインクリプションオラクル:** Query Phase 1 と同様. 但し,  $(C, A_s, S_r) = (C^*, A_s^*, S_r(s.t. S_r \in A_d^*))$  のクエリは禁止.

**Guess Phase:**  $\mathcal{A}$  はチャレンジビット  $b$  に対する推測として  $b' \in \{0, 1\}$  を出力.

KP-ABSC 方式  $\Pi_{\text{KSC}}$  に関する AA-IND-CCA 安全性ゲームにおいて攻撃者  $\mathcal{A}$  の優位性を次式で定義する.

$$\text{Adv}_{\Pi_{\text{KSC}}, \mathcal{A}}^{\text{AA-IND-CCA}} = |\Pr[b' = b] - \frac{1}{2}| \quad (2.41)$$

**定義 2.35.** 全ての PPTA 攻撃者  $\mathcal{A}$  に対して, KP-ABSC 方式  $\Pi_{\text{KSC}}$  に関する AA-IND-CCA 安全性ゲームにおける  $\mathcal{A}$  の優位性  $\text{Adv}_{\Pi_{\text{KSC}}, \mathcal{A}}^{\text{AA-IND-CCA}}$  が, セキュリティパラメータ  $k$  に関して無視できるほど小さい値であるならば, KP-ABSC 方式  $\Pi_{\text{KSC}}$  は AA-IND-CCA 安全である.

**AA-IND-CCA 安全性は“内部者安全性”である.**

KP-ABSC の AA-IND-CCA 安全性が内部者安全性であることは, CP-ABSC の AP-IND-CCA 安全性が内部者安全性であること (2.8.1 項を参照) と同様の理由によるため, 詳細な説明は割愛する.

## 2.9.2 SA-IND-CCA 安全性

選択的属性モデルにおける適応的選択暗号文(サインクリプテキスト)攻撃に対する暗号文(サインクリプテキスト)識別不可能性 (Ciphertext(Signcryptext)-Indistinguishability under Adaptively Chosen Ciphertext(Signcryptext) Attacks in the Selective Attribute Model, SA-IND-CCA) は, KP-ABSC 方式が満たすべき秘匿性に関する安全性の中で, AA-IND-CCA(2.9.1) より弱いことが自明な安全性である. KP-ABSC 方式  $\Pi_{\text{KSC}}$  に関する SA-IND-CCA 安全性は, 攻撃者  $\mathcal{A}$  と挑戦者  $\mathcal{CH}$  との間で行われる以下の SA-IND-CCA 安全性ゲームによって定義する.

**Init Phase:**  $\mathcal{CH}$  はターゲット復号者属性集合  $S_d^*$  を  $\mathcal{CH}$  へ送る.

**Setup Phase:**  $\mathcal{CH}$  が  $\text{KSC.Setup}(1^k, \mathcal{U}_s, \mathcal{U}_r) \rightarrow (\text{PK}, \text{MK})$  を実行し, システム公開鍵 PK を  $\mathcal{A}$  に渡す.

**Query Phase 1:**  $\mathcal{A}$  は以下の各オラクルに対してクエリを任意回数発行することができる

**送信者用鍵生成:**  $\mathcal{A}$  は送信者のポリシー  $\mathbb{A}_s$  をクエリする.  $\mathcal{CH}$  は  $\text{KSC.KeyGen}_s(\text{PK}, \text{MK}, \mathbb{A}_s) \rightarrow \text{SK}_s$  を実行し  $\text{SK}_s$  を  $\mathcal{CH}$  へ送る.

**受信者用鍵生成:**  $\mathcal{A}$  は受信者のポリシー  $\mathbb{A}_r$  をクエリする. 但し,  $S_d^* \in \mathbb{A}_r$  を満たす  $\mathbb{A}_r$  のクエリは禁止.  $\mathcal{CH}$  は  $\text{KeyGen}_R(\text{PK}, \text{MK}, \mathbb{A}_r) \rightarrow \text{SK}_r$  を実行し  $\text{SK}_r$  を  $\mathcal{CH}$  へ送る.

**サインクリプションオラクル:**  $\mathcal{A}$  は平文  $m \in \mathcal{M}$ , 署名者属性集合  $S_s$ , 復号者属性集合  $S_d$ ,  $S_s \in \mathbb{A}_s$  を満たすポリシー  $\mathbb{A}_s$  をクエリする.  $\mathcal{CH}$  は  $\text{KSC.KeyGen}_s(\text{PK}, \text{MK}, \mathbb{A}_s) \rightarrow \text{SK}_s$ ,  $\text{KSC.SC}(\text{PK}, m, \text{SK}_s, S_s, S_d) \rightarrow C$  を実行し,  $C$  を  $\mathcal{A}$  へ送る.

**アンサインクリプションオラクル:**  $\mathcal{A}$  はサインクリプトテキスト  $C \in \mathcal{C}$ , 署名者属性集合  $S_s$ , ポリシー  $\mathbb{A}_r$  をクエリする.  $\mathcal{CH}$  は  $\text{KSC.KeyGen}_R(\text{PK}, \text{MK}, \mathbb{A}_r) \rightarrow \text{SK}_r$ ,  $\text{KSC.USC}(\text{PK}, C, \text{SK}_r, S_s) \rightarrow m/\perp$  を実行し最終実行結果を  $\mathcal{A}$  へ送る.

**Challenge Phase:**  $\mathcal{A}$  は復号者属性集合  $S_d^*$ , 署名者属性集合  $S_s^*$ ,  $S_s^* \in \mathbb{A}_s$  を満たすポリシー  $\mathbb{A}_s$ , 及びサイズの等しい二つの平文  $m_0 \in \mathcal{M}, m_1 \in \mathcal{M}$  を選び  $\mathcal{CH}$  に送る.  $\mathcal{CH}$  は  $\text{SK}_s^* \leftarrow \text{KSC.KeyGen}_s(\text{PK}, \text{MK}, \mathbb{A}_s), b \xleftarrow{\text{U}} \{0, 1\}, C^* \leftarrow \text{KSC.SC}(\text{PK}, m_b^*, \text{SK}_s^*, S_s^*, S_d^*)$  を実行し, チャレンジサインクリプトテキスト  $C^*$  を  $\mathcal{A}$  に送る.

**Query Phase 2:**  $\mathcal{A}$  は以下の各オラクルに対してクエリを任意回数発行することができる.

**送信者用鍵生成:** Query Phase 1 と同様.

**受信者用鍵生成:** Query Phase 1 と同様.

**サインクリプションオラクル:** Query Phase 1 と同様.

**アンサインクリプションオラクル:** Query Phase 1 と同様. 但し,  $(C, S_s, \mathbb{A}_r) = (C^*, S_s^*, \mathbb{A}_r(s.t. S_d^* \in \mathbb{A}_r))$  のクエリは禁止.

**Guess Phase:**  $\mathcal{A}$  はチャレンジビット  $b$  に対する推測として  $b' \in \{0, 1\}$  を出力.

KP-ABSC 方式  $\Pi_{\text{KSC}}$  に関する SA-IND-CCA 安全性ゲームにおいて攻撃者  $\mathcal{A}$  の優位性を次式で定義する.

$$\text{Adv}_{\Pi_{\text{KSC}}, \mathcal{A}}^{\text{SA-IND-CCA}} = |\Pr[b' = b] - \frac{1}{2}| \quad (2.42)$$

**定義 2.36.** 全ての PPTA 攻撃者  $\mathcal{A}$  に対して, KP-ABSC 方式  $\Pi_{\text{KSC}}$  に関する SA-IND-CCA 安全性ゲームにおける  $\mathcal{A}$  の優位性  $\text{Adv}_{\Pi_{\text{KSC}}, \mathcal{A}}^{\text{SA-IND-CCA}}$  が, セキュリティパラメータ  $k$  に関して無視できるほど小さい値であるならば, KP-ABSC 方式  $\Pi_{\text{KSC}}$  は SA-IND-CCA 安全である.

### 2.9.3 AA-sEUF-CMA 安全性

適応的属性モデルにおける適応的選択文書攻撃に対するサインクリプテキスト強偽造不可能性 (Strongly Existentially Unforgeability of Signcryptext under Adaptively Chosen Message Attacks in the Adaptive Attribute Model, AA-sEUF-CMA) は, KP-ABSC 方式が満たすべき完全性に関する安全性の中で, 最も強い安全性である. KP-ABSC 方式  $\Pi_{\text{KSC}}$  に関する AA-sEUF-CMA 安全性は, 攻撃者  $\mathcal{A}$  と挑戦者  $\mathcal{CH}$  との間で行われる以下の AA-sEUF-CMA 安全性ゲームによって定義する.

**Setup Phase:**  $\mathcal{CH}$  が  $\text{KSC.Setup}(1^k, \mathcal{U}_s, \mathcal{U}_r) \rightarrow (\text{PK}, \text{MK})$  を実行し, システム公開鍵  $\text{PK}$  を  $\mathcal{A}$  に渡す.

**Query Phase:**  $\mathcal{A}$  は以下の各オラクルに対してクエリを任意回数発行することができる.

**送信者用鍵生成:**  $\mathcal{A}$  はポリシー  $\mathbb{A}_s$  をクエリする.  $\mathcal{CH}$  は  $\text{KSC.KeyGen}_s(\text{PK}, \text{MK}, \mathbb{A}_s) \rightarrow \text{SK}_s$  を実行し  $\text{SK}_s$  を  $\mathcal{A}$  へ送る. 但し,  $S_s^* \in \mathbb{A}_s$  を満たす  $\mathbb{A}_s$  をクエリすることは禁止とする.

**受信者用鍵生成:**  $\mathcal{A}$  はポリシー  $\mathbb{A}_r$  をクエリする.  $\mathcal{CH}$  は  $\text{KSC.KeyGen}_r(\text{PK}, \text{MK}, \mathbb{A}_r) \rightarrow \text{SK}_r$  を実行し  $\text{SK}_r$  を  $\mathcal{A}$  へ送る.

**サインクリプション:**  $\mathcal{A}$  は平文  $m \in \mathcal{M}$ , 署名者属性集合  $S_s$ , 復号者属性集合  $S_d$ ,  $S_s \in \mathbb{A}_s$  を満たすポリシー  $\mathbb{A}_s$  を選び,  $\mathcal{CH}$  へ送る.  $\mathcal{CH}$  は  $\text{KSC.KeyGen}_s(\text{PK}, \text{MK}, \mathbb{A}_s) \rightarrow \text{SK}_s$ ,  $\text{KSC.SC}(\text{PK}, m, \text{SK}_s, S_s, S_d) \rightarrow C$  を実行し,  $C$  を  $\mathcal{A}$  へ送る.  $\mathcal{CH}$  は  $(m, C, S_s, S_d)$  を  $\mathcal{L}_{\text{KSC.SC}}$  へ追加する.

**アンサインクリプション:**  $\mathcal{A}$  はサインクリプテキスト  $C \in \mathcal{C}$ , 署名者属性集合  $S_s$ , ポリシー  $\mathbb{A}_r$  を  $\mathcal{CH}$  へ送る.  $\mathcal{CH}$  は  $\text{KSC.KeyGen}_r(\text{PK}, \text{MK}, \mathbb{A}_r) \rightarrow \text{SK}_r$ ,  $\text{KSC.SC}(\text{PK}, C, \text{SK}_r, S_s) \rightarrow m/\perp$  を実行し最後の実行結果を  $\mathcal{A}$  へ送る.

**Forgery:**  $\mathcal{A}$  が  $(C^*, S_s^*, S_d^*)$  を出力.

KP-ABSC 方式  $\Pi_{\text{KSC}}$  に関する AA-sEUF-CMA 安全性ゲームにおいて攻撃者  $\mathcal{A}$  の優位性を次式で定義する. 式 (2.43) 中,  $n$  は  $S_d^* \in \mathbb{A}_r$  を満たす  $\mathbb{A}_r$  の個数を表す. また, 同

式中,  $\text{Disclose}_{\text{KSC}}$  は復号者属性集合開示性アルゴリズムを表す.

$$\begin{aligned}
& \text{Adv}_{\Pi_{\text{KSC}}, \mathcal{A}}^{\text{AA-sEUF-CMA}} \\
= & \Pr[[\text{Disclose}_{\text{KSC}}(C^*) = S_d^*] \\
& \wedge [\forall \mathbb{A}_r^{(i)} \ni S_d^*, \text{KSC.KeyGen}_R(\text{PK}, \text{MK}, \mathbb{A}_r^{(i)}) \rightarrow \text{SK}_r^{(i)}, \\
& \quad \text{KSC.USC}(\text{PK}, C^*, \text{SK}_r^{(i)}, S_s^*) \rightarrow m^{(i)} \in \mathcal{M}] \\
& \wedge [m^{(1)} = \dots = m^{(n)} =: m^*] \wedge \underline{[(m^*, C^*, S_s^*, S_d^*) \notin \mathcal{L}_{\text{KSC.sc}}]}] \quad (2.43)
\end{aligned}$$

**定義 2.37.** 全ての  $PPTA$  攻撃者  $\mathcal{A}$  に対して,  $KP\text{-}ABSC$  方式  $\Pi_{\text{KSC}}$  に関する  $AA\text{-}sEUF\text{-}CMA$  安全性ゲームにおける  $\mathcal{A}$  の優位性  $\text{Adv}_{\Pi_{\text{KSC}}, \mathcal{A}}^{\text{AA-sEUF-CMA}}$  が, セキュリティパラメータ  $k$  に関して無視できるほど小さい値であるならば,  $KP\text{-}ABSC$  方式  $\Pi_{\text{KSC}}$  は  $AA\text{-}sEUF\text{-}CMA$  安全である.

定義 2.37 は, “強偽造不可能性” である. それに対して, “弱偽造不可能性”, 正式には, 適応的属性モデルにおける適応的選択文書攻撃に対するサインクリプトテキスト弱偽造不可能性 (Weakly Existentially Unforgeability of Signcryptext under Adaptively Chosen Message Attacks in the Adaptive Attribute Model,  $AA\text{-}wEUF\text{-}CMA$ ) は, 先の  $AA\text{-}sEUF\text{-}CMA$  安全性ゲームと同一の  $AA\text{-}wEUF\text{-}CMA$  安全性ゲームにおいて, 攻撃者の優位性を次式で定義する. ここで, 式 (2.43) と式 (2.44) の違いは, 下線部分である.

$$\begin{aligned}
& \text{Adv}_{\Pi_{\text{KSC}}, \mathcal{A}}^{\text{AA-wEUF-CMA}} \\
= & \Pr[[\text{Disclose}_{\text{KSC}}(C^*) = S_d^*] \\
& \wedge [\forall \mathbb{A}_r^{(i)} \ni S_d^*, \text{KSC.KeyGen}_R(\text{PK}, \text{MK}, \mathbb{A}_r^{(i)}) \rightarrow \text{SK}_r^{(i)}, \\
& \quad \text{KSC.USC}(\text{PK}, C^*, \text{SK}_r^{(i)}, S_s^*) \rightarrow m^{(i)} \in \mathcal{M}] \\
& \wedge [m^{(1)} = \dots = m^{(n)} =: m^*] \wedge \underline{[(m^*, S_s^*, S_d^*) \notin \mathcal{L}_{\text{KSC.sc}}]}] \quad (2.44)
\end{aligned}$$

**定義 2.38.** 全ての  $PPTA$  攻撃者  $\mathcal{A}$  に対して,  $KP\text{-}ABSC$  方式  $\Pi_{\text{KSC}}$  に関する  $AA\text{-}wEUF\text{-}CMA$  安全性ゲームにおける  $\mathcal{A}$  の優位性  $\text{Adv}_{\Pi_{\text{KSC}}, \mathcal{A}}^{\text{AA-wEUF-CMA}}$  が, セキュリティパラメータ  $k$  に関して無視できるほど小さい値であるならば,  $KP\text{-}ABSC$  方式  $\Pi_{\text{KSC}}$  は  $AA\text{-}wEUF\text{-}CMA$  安全である.

## 2.9.4 SA-sEUF-CMA 安全性

選択的属性モデルにおける適応的選択文書攻撃に対するサインクリプトテキスト強偽造不可能性 (Strongly Existentially Unforgeability of Signcryptext under Adaptively Chosen Message Attacks in the Selective Attribute Model,  $SA\text{-}sEUF\text{-}CMA$ ) は,  $KP\text{-}ABSC$  方式が満たすべき完全性に関する安全性の中で,  $AA\text{-}sEUF\text{-}CMA$  (2.9.3 項) よりも弱いことが自明な安全性である.  $KP\text{-}ABSC$  方式  $\Pi_{\text{KSC}}$  に関する  $SA\text{-}sEUF\text{-}CMA$  安全性は, 攻撃者  $\mathcal{A}$  と挑戦者  $\mathcal{CH}$  との間で行われる以下の  $SA\text{-}sEUF\text{-}CMA$  安全性ゲームによって定義する.

**Init Phase:**  $\mathcal{CH}$  はターゲット復号者属性集合  $S_d^*$  を  $\mathcal{CH}$  へ送る.

**Setup Phase:**  $\mathcal{CH}$  が  $\text{KSC.Setup}(1^k, \mathcal{U}_s, \mathcal{U}_r) \rightarrow (\text{PK}, \text{MK})$  を実行し, システム公開鍵  $\text{PK}$  を  $\mathcal{A}$  に渡す.

**Query Phase:**  $\mathcal{A}$  は以下の各オラクルに対してクエリを任意回数発行することができる.

**送信者用鍵生成:**  $\mathcal{A}$  はポリシー  $\mathbb{A}_s$  をクエリする.  $\mathcal{CH}$  は  $\text{KSC.KeyGen}_s(\text{PK}, \text{MK}, \mathbb{A}_s) \rightarrow \text{SK}_s$  を実行し  $\text{SK}_s$  を  $\mathcal{A}$  へ送る. 但し,  $S_s^* \in \mathbb{A}_s$  を満たす  $\mathbb{A}_s$  をクエリすることは禁止とする.

**受信者用鍵生成:**  $\mathcal{A}$  はポリシー  $\mathbb{A}_r$  をクエリする.  $\mathcal{CH}$  は  $\text{KSC.KeyGen}_r(\text{PK}, \text{MK}, \mathbb{A}_r) \rightarrow \text{SK}_r$  を実行し  $\text{SK}_r$  を  $\mathcal{A}$  へ送る.

**サインクリプション:**  $\mathcal{A}$  は平文  $m \in \mathcal{M}$ , 署名者属性集合  $S_s$ , 復号者属性集合  $S_d$ ,  $S_s \in \mathbb{A}_s$  を満たすポリシー  $\mathbb{A}_s$  を選び,  $\mathcal{CH}$  へ送る.  $\mathcal{CH}$  は  $\text{KSC.KeyGen}_s(\text{PK}, \text{MK}, \mathbb{A}_s) \rightarrow \text{SK}_s$ ,  $\text{KSC.SC}(\text{PK}, m, \text{SK}_s, S_s, S_d) \rightarrow C$  を実行し,  $C$  を  $\mathcal{A}$  へ送る.  $\mathcal{CH}$  は  $(m, C, S_s, S_d)$  を  $\mathcal{L}_{\text{KSC.SC}}$  へ追加する.

**アンサインクリプション:**  $\mathcal{A}$  はサインクリプテキスト  $C \in \mathcal{C}$ , 署名者属性集合  $S_s$ , ポリシー  $\mathbb{A}_r$  を  $\mathcal{CH}$  へ送る.  $\mathcal{CH}$  は  $\text{KSC.KeyGen}_r(\text{PK}, \text{MK}, \mathbb{A}_r) \rightarrow \text{SK}_r$ ,  $\text{KSC.SC}(\text{PK}, C, \text{SK}_r, S_s) \rightarrow m/\perp$  を実行し最後の実行結果を  $\mathcal{A}$  へ送る.

**Forgery:**  $\mathcal{A}$  が  $(C^*, S_s^*)$  を出力.

KP-ABSC 方式  $\Pi_{\text{KSC}}$  に関する SA-sEUF-CMA 安全性ゲームにおいて攻撃者  $\mathcal{A}$  の優位性を次式で定義する. 式 (??) 中,  $n$  は  $S_d^* \in \mathbb{A}_r$  を満たす  $\mathbb{A}_r$  の個数を表す. また, 同式中,  $\text{KSC}$  は, KP-ABSC 方式  $\Pi_{\text{KSC}}$  の復号者属性集合開示性アルゴリズムである.

$$\begin{aligned}
& \text{Adv}_{\Pi_{\text{KSC}}, \mathcal{A}}^{\text{SA-sEUF-CMA}} \\
&= \Pr[\text{Disclose}_{\text{KSC}}(C^*) = S_d^*] \\
& \quad \wedge [\forall S_d^* \in \mathbb{A}_r^{(i)}, \text{KSC.KeyGen}_r(\text{PK}, \text{MK}, \mathbb{A}_r^{(i)}) \rightarrow \text{SK}_r^{(i)}, \\
& \quad \quad \text{KSC.USC}(\text{PK}, C^*, \text{SK}_r^{(i)}, S_s^*) \rightarrow m^{(i)} \in \mathcal{M}] \\
& \quad \wedge [m^{(1)} = \dots = m^{(n)} =: m^*] \wedge [(m^*, C^*, S_s^*, S_d^*) \notin \mathcal{L}_{\text{KSC.SC}}] \quad (2.45)
\end{aligned}$$

**定義 2.39.** 全ての PPTA 攻撃者  $\mathcal{A}$  に対して, KP-ABSC 方式  $\Pi_{\text{KSC}}$  に関する SA-sEUF-CMA 安全性ゲームにおける  $\mathcal{A}$  の優位性  $\text{Adv}_{\Pi_{\text{KSC}}, \mathcal{A}}^{\text{SA-sEUF-CMA}}$  が, セキュリティパラメータ  $k$  に関して無視できるほど小さい値であるならば, KP-ABSC 方式  $\Pi_{\text{KSC}}$  は SA-sEUF-CMA 安全である.

定義 2.39 は, “強偽造不可能性”である. それに対して, “弱偽造不可能性”, 正式には, 選択的属性モデルにおける適応的選択文書攻撃に対するサインクリプテキスト弱偽造不可能性 (Weakly Existentially Unforgeability of Signcryptext under Adaptively Chosen Message Attacks in the Selective Attribute Model, SA-wEUF-CMA) は, 先の SA-sEUF-CMA 安全性ゲームと同一の SA-wEUF-CMA 安全性ゲームにおいて, 攻撃者の優位性を次式で定義する. ここで, 式 (2.45) と式 (2.46) の違いは, 下線部分である.

$$\begin{aligned}
& \text{Adv}_{\Pi_{\text{KSC}}, \mathcal{A}}^{\text{SA-wEUF-CMA}} \\
= & \Pr[[\text{Disclose}_{\text{KSC}}(C^*) = S_d^*] \\
& \wedge [\forall S_d^* \in \mathbb{A}_r^{(i)}, \text{KSC.KeyGen}_R(\text{PK}, \text{MK}, \mathbb{A}_r^{(i)}) \rightarrow \text{SK}_r^{(i)}, \\
& \quad \text{KSC.USC}(\text{PK}, C^*, \text{SK}_r^{(i)}, S_s^*) \rightarrow m^{(i)} \in \mathcal{M}] \\
& \wedge [m^{(1)} = \dots = m^{(n)} =: m^*] \wedge \underline{[(m^*, S_s^*, S_d^*) \notin \mathcal{L}_{\text{KSC.sc}}]}] \quad (2.46)
\end{aligned}$$

**定義 2.40.** 全ての PPTA 攻撃者  $\mathcal{A}$  に対して, KP-ABSC 方式  $\Pi_{\text{KSC}}$  に関する SA-wEUF-CMA 安全性ゲームにおける  $\mathcal{A}$  の優位性  $\text{Adv}_{\Pi_{\text{KSC}}, \mathcal{A}}^{\text{SA-wEUF-CMA}}$  が, セキュリティパラメータ  $k$  に関して無視できるほど小さい値であるならば, KP-ABSC 方式  $\Pi_{\text{KSC}}$  は SA-wEUF-CMA 安全である.

## 2.9.5 完全匿名性

CP-ABSC 方式の完全匿名性 (Perfect Privacy, あるいは Signer Privacy) は, 直感的な説明としては, サインクリプテキストからサインクリプタの属性集合が漏れないことを保障する安全性である. 具体的には, 次のように定義される.

**定義 2.41.** CP-ABSC 方式が完全匿名性を満たすとは, 当該方式が以下の条件を満たす場合を言う. :

全ての  $(\text{PK}, \text{MK}) \leftarrow \text{KSC.Setup}(1^k, \mathcal{U}_s, \mathcal{U}_r)$ , 全ての  $\mathbb{A}_s$ , 全ての  $\mathbb{A}'_s$ , 全ての  $\text{SK}_s \leftarrow \text{KSC.KeyGen}_s(\text{PK}, \text{MK}, \mathbb{A}_s)$ , 全ての  $\text{SK}'_s \leftarrow \text{KSC.KeyGen}_s(\text{PK}, \text{MK}, \mathbb{A}'_s)$ , 全ての  $m \in \mathcal{M}$ , 全ての  $S_s (s.t. S_s \in \mathbb{A}_s \wedge S_s \in \mathbb{A}'_s)$ , 全ての  $S_d$  に対して,  $\text{KSC.SC}(\text{PK}, m, \text{SK}_s, S_s, S_d)$  の確率分布と  $\text{KSC.SC}(\text{PK}, m, \text{SK}'_s, S_s, S_d)$  の確率分布が同一である.

## 2.9.6 復号者属性集合開示性

KP-ABSC 方式の “復号者属性集合開示性” は著者が独自に命名し, かつ定義した性質である. 直感的には, サインクリプテキストからサインクリプテキスト生成に使用された (あるいは, サインクリプテキストと真に関連付けられた) 復号者属性集合を完璧に復元できることを意味する. 具体的には, 次の様に定義される.

**定義 2.42.** KP-ABSC 方式が復号者属性集合開示性を満たすとは, PPTA  $\text{Disclose}_{\text{KSC}}$  が存在し, 全ての  $k$ , 全ての  $\mathcal{U}_s$ , 全ての  $\mathcal{U}_r$ , 全ての  $(\text{PK}, \text{MK}) \leftarrow \text{KSC.Setup}(1^k, \mathcal{U}_s, \mathcal{U}_r)$ , 全

での  $m$ , 全ての  $A_s$ , 全ての  $SK_s \leftarrow \text{KSC.KeyGen}_s(\text{PK}, \text{MK}, A_s)$ , 全ての  $S_s (s.t. S_s \in A_s)$ , 全ての  $S_d$ , 全ての  $C \leftarrow \text{KS.SC}(\text{PK}, m, SK_s, S_s, S_d)$  に対して, 以下の条件式が成立する場合を言う.

$$\Pr[S'_d = S_d | S'_d \leftarrow \text{Disclose}_{\text{KSC}}(\text{PK}, C)] = 1 \quad (2.47)$$

KP-ABSC 方式の“復号者属性集合開示性”は特殊な性質ではなく極めて自然な性質である. 事実として著者が知る限り, 既存の KP-ABSC 方式のうちの多く [6][9] に関して, 容易に定義 2.42 の条件を満たすような PPTA  $\text{Disclose}_{\text{KSC}}$  を構成できる.

## Chapter 3 属性ベース Signcryption の関連研究

属性ベース Signcryption の関連研究を表 3.1 に示す。

Pandit ら [4][5] は、Combined Setup 型<sup>1</sup>の CP-ABSC 方式の構成法を提案している。安全性としては、CP-ABSC が達成すべき安全性の中で最も望ましいとされている安全性 (AP-IND-CCA, AP-sEUF-CMA, 完全匿名性) を達成可能である。仮定としては、標準的な仮定と言われている DLIN 仮定 (Decisional Linear Assumption) と比べると、より非標準的 (より強い) と言われている Decisional Sub-Group 仮定が 3 種類必要である。

Chen ら [3] は、Combined Setup 型の CP-ABSC の構成法を提案している。Chen らの CP-ABSC の構成法が達成可能な安全性は、秘匿性 (Confidentiality) に関しては SP-IND-CCA, 偽造不可能性 (Unforgeability) に関しては SP-wEUF-CMA であり、いずれも最強の安全性ではない。Chen らは、“Combined CP-ABE and ABS (CCP-ABES)” という要素技術を独自に定義し、CP-ABSC の構成法の安全性を示すための仮定として利用している。CCP-ABES は、一つの秘密鍵で、CP-ABE と SP-ABS の両機能を実現する要素技術であり、セットアップ、秘密鍵生成、暗号化、復号、署名生成、署名検証の 6 つのアルゴリズムを持つ。CCP-ABES の秘匿性に関しては、“IND-CCA security in the presence of a signing oracle” という安全性を定義しており、これは一般的な CP-ABE の IND-CCA の定義に署名生成オラクルが追加されたものとみなせる。以降当該安全性は「署名オラクル付き IND-CCA 安全性」と呼ぶ。署名オラクル付き IND-CCA 安全性は、一般的な CP-ABE の IND-CCA 定義と同様に、復号者アクセス構造を攻撃者が指定するタイミングにより、適応的安全性と選択的安全性が存在し、以降では前者を「署名オラクル付き AP-IND-CCA」、後者を「署名オラクル付き SP-IND-CCA」と呼ぶ。偽造不可能性に関しては、“EUF-CMA security in the presence of a decryption oracle” という安全性を定義しており、IND-CCA と同様に、これも一般的な SP-ABS の wEUF-CMA 定義に復号オラクルが追加されたものとみなせる。以降当該安全性は「復号オラクル付き wEUF-CMA 安全性」と呼ぶ。また、IND-CCA と同様に、これにも適応的安全性と選択的安全性が存在し、前者を「復号オラクル付き AP-wEUF-CMA」、後者を「復号オラクル付き SP-wEUF-CMA」と呼ぶ。Chen らは CP-ABSC の構成法の SP-IND-CCA の安全性証明において、「署名オラクル付き SP-IND-CCA 安全な CCP-ABES」を仮定として利用している。しかし、SP-IND-CCA 安全な CP-ABE から、署名オラクル付き

---

<sup>1</sup>Combined Setup 型の CP-ABSC とは、サインクリプション実行に用いる秘密鍵の生成と、アンサインクリプション実行に用いる秘密鍵の生成において、単一の共通のアルゴリズムが用いられるような構造の CP-ABSC である。ユーザ視点で見れば、各ユーザが管理すべき秘密鍵が一つだけで済み、その秘密鍵でサインクリプションとアンサインクリプションの両アルゴリズムが実行できる CP-ABSC である。サインクリプションとアンサインクリプションの各アルゴリズムの実行に異なる秘密鍵が必要な非 Combined Setup 型の CP-ABSC と比べるとユーザの鍵管理の負担がより少ないため実用性はその分高いと言える。

SP-IND-CCA 安全な CCP-ABES を構成できるかどうかは非自明であり, [3] にはその証明はない. また, Chen らは CP-ABSC の構成法の SP-wEUF-CMA の安全性証明において, 「復号オラクル付き SP-wEUF-CMA 安全な CCP-ABES」を仮定として利用している. しかし, 秘匿性と同様に, SP-wEUF-CMA 安全な SP-ABS から, 復号オラクル付き SP-wEUF-CMA 安全な CCP-ABES を構成できるかどうかは非自明であり, [3] にはその証明はない. 従って, Chen らの CP-ABSC の構成法は, (少なくとも現時点では) 安全性を CP-ABE と SP-ABS の安全性に直接的に帰着するものではない.

Rao ら [6] は, 暗号文サイズ, サインクリプションコスト, アンサインクリプションコストが一定である KP-ABSC の構成法を提案している. しかし, Rao らの構成法が達成可能な安全性は, 秘匿性は SA-IND-CCA, 偽造不可能性は SA-sEUF-CMA であり, いずれも最強の安全性ではない.

Datta ら [7][8] は, ポリシーとして任意の多項式サイズの回路を使用できる KP-ABSC と CP-ABSC の構成法を提案している. Datta らの構成法は, KP 型と CP 型いずれにおいても, 識別不可能性難読化器と Statistically Simulation-Sound Non-Interactive Zero Knowledge Proof of Knowledge (SSS-NIZKPoK) を利用しているが, これらは非常に仮定が強いとされている暗号技術である. また, 達成できる安全性は非常に弱い. KP-ABSC (resp. CP-ABSC) が達成できる秘匿性は, 通常の SA-IND-CPA (resp. SP-IND-CPA) の定義と比べると攻撃者に対してより厳しい (より不利な) 条件を課した定義での秘匿性であり, つまり SA-IND-CPA (resp. SP-IND-CPA) という非常に弱いとされている秘匿性よりも更に弱い. 一方で, Datta らの KP-ABSC (resp. CP-ABSC) が達成できる偽造不可能性は, 秘匿性と同様に, 通常の SA-wUUF-CMA (resp. SP-wUUF-CMA) の定義と比べると攻撃者に対してより厳しい条件を課した定義での偽造不可能性であり, つまり SA-wUUF-CMA (resp. SP-wUUF-CMA) という非常に弱いとされている偽造不可能性よりも更に弱い.

Nandi ら [9] は, 述語 Signcryption (Predicate Signcryption) の一般的構成法を提案している. ここで, 述語暗号 (Predicate Encryption) は, CP-ABE と KP-ABE 等の暗号技術の一般化であり, 述語関数と呼ばれる関数によって, CP-ABE の機能が実現されたり, KP-ABE の機能が実現されたりする暗号技術である. それに対して, 述語 Signcryption は, CP-ABSC と KP-ABSC 等の暗号技術の一般化である. Nandi らの述語 Signcryption は, Attrapadung [10][11] が提案した Pair Encoding と呼ばれる暗号技術を構成要素とした構成法であり, CP-ABSC (resp. KP-ABSC) の場合, 安全性として, AP-IND-CCA (resp. AA-IND-CCA), AP-sEUF-CMA (resp. AA-sEUF-CMA), 完全匿名性を達成可能である. 安全性仮定は, 特定の安全性及び性質を満たす Pair Encoding, 3 種類の Decisional Sub-Group 仮定などである.

以上より, これまでに, CP-ABE (あるいは, CP-ABKEM), SP-ABS を構成要素とした CP-ABSC の一般的構成法を提案し, 安全性を CP-ABE (CP-ABKEM), SP-ABS の安全性に直接的に帰着させ, かつ CP-ABSC が満たすべき安全性の中の最も強い安全性である AP-IND-CCA, AP-sEUF-CMA, 完全匿名性を達成可能であることを示した既存研究は存在しない. また, KP-ABSC に関しても同様に, KP-ABE (KP-ABKEM), KP-ABS 等を構成要素とした KP-ABSC の一般的構成法を提案し, 安全性を KP-ABE (KP-

ABKEM), KP-ABS の安全性に直接的に帰着させ, かつ最も強い安全性である, AA-IND-CCA, AA-sEUF-CMA, 完全匿名性を達成可能であることを示した研究はこれまでに行われていない.

方式	CP /KP	Confidentiality /Unforgeability	PP?	仮定	CS?
Pandit [4][5]	CP	AP-IND-CCA /AP-sEUF-CMA	Yes	衝突困難なハッシュ関数, 秘匿性 (hiding property) を持つコミットメント, sEUF-CMA 安全なワンタイム署名, 3種類の Decisional Sub-Group 仮定.	Yes
Chen [3]	CP	SP-IND-CCA /SP-wEUF-CMA	Yes	署名オラクル付き SP-IND-CCA 安全, かつ復号オラクル付き SP-wEUF-CMA 安全, かつ完全匿名な CCP-ABES.	Yes
Rao[6]	KP	SA-IND-CCA /SA-sEUF-CMA	(論文内で未記述の為不明.)	衝突困難なハッシュ関数, dBDHE, cDHE.	No
Datta [7][8]	KP	SA-IND-CPA よりも弱い/SA-wUUF-CMA よりも弱い	(論文内で未記述の為不明.)	IND-CPA 安全な公開鍵暗号, wEUF-CMA 安全な署名, 識別不可能性難読化器, SSS-NIZKPoK.	No
Datta [7][8]	CP	SP-IND-CPA よりも弱い/SP-wUUF-CMA よりも弱い	(論文内で未記述の為不明.)	IND-CPA 安全な公開鍵暗号, wEUF-CMA 安全な署名, 識別不可能性難読化器, SSS-NIZKPoK.	No
Nandi [9]	CP	AP-IND-CCA /AP-sEUF-CMA	Yes	衝突困難なハッシュ関数, 秘匿性 (hiding property) を持つコミットメント, sEUF-CMA 安全なワンタイム署名, (特定の安全性・条件を満たす) Pair Encoding[10][11], 3種類の Decisional Sub-Group 仮定.	Yes
Nandi [9]	KP	AA-IND-CCA /AA-sEUF-CMA	Yes	衝突困難なハッシュ関数, 秘匿性 (hiding property) を持つコミットメント, sEUF-CMA 安全なワンタイム署名, (特定の安全性・条件を満たす) Pair Encoding[10][11], 3種類の Decisional Sub-Group 仮定.	Yes

表 3.1: 属性ベース Signcryption の関連研究 (表中の略語の意味は以下の通りである. PP: 完全匿名性 (Perfect Privacy), CS: Combined Setup)

# Chapter 4 暗号文ポリシー型属性ベース鍵カプセル化メカニズムの一般的構成

## 4.1 本章の概要

本章に示す成果は、暗号文ポリシー型属性ベース暗号 (CP-ABE) を構成要素とした、暗号文ポリシー型属性ベース鍵カプセル化メカニズム (CP-ABKEM) の一般的構成法とその安全性証明である。具体的には、AP-IND-CCA 安全、かつ復号者アクセス構造開示的な CP-ABE から、AP-IND-CCA 安全、かつ復号者アクセス構造開示的な CP-ABKEM を一般的に構成できることを証明した。

なお、本成果 (暗号文ポリシー型属性ベース鍵カプセル化メカニズムの一般的構成) の意義については、8.1.1 項で説明する。

また、本章の構成については以下の通りである。4.2 節で一般的構成法の説明を行う。4.3 節で安全性証明の詳述を行う。

## 4.2 提案する一般的構成法

構成要素として、CP-ABE 方式  $\Pi_{CE} : (\text{CE.Setup}, \text{CE.KeyGen}, \text{CE.Enc}, \text{CE.Dec})$  を用いた、CP-ABKEM 方式  $\Pi_{CK} : (\text{CK.Setup}, \text{CK.KeyGen}, \text{CK.Encap}, \text{CK.Decap})$  の一般的構成法を図 4.1 に示す。

CK.Setup( $1^k, \mathcal{U}$ ) : Return $(\text{PK}, \text{MK}) \leftarrow \text{CE.Setup}(1^k, \mathcal{U})$ .
CK.KeyGen( $\text{PK}, \text{MK}, S$ ) : Return $\text{SK} \leftarrow \text{CE.KeyGen}(\text{PK}, \text{MK}, S)$ .
CK.Encap( $\text{PK}, \mathbb{A}_d$ ) : $K \xleftarrow{\mathcal{U}} \mathcal{K}; C_K \leftarrow \text{CE.Enc}(\text{PK}, K, \mathbb{A}_d);$ Return $(K, C_K)$ .
CK.Decap( $\text{PK}, C_K, \text{SK}$ ) : $\alpha := \text{CE.Dec}(\text{PK}, C_K, \text{SK}_r);$ If $\alpha = \perp$ , then return $\perp$ . Else return $K := \alpha$ .

図 4.1: CP-ABKEM 方式の一般的構成法  $\Pi_{CK}$

### 4.3 安全性証明

図 4.1 の CP-ABKEM 方式の一般的構成法  $\Pi_{\text{CK}}$  に関して, 定理 4.1, 定理 4.2 が成立する.

**定理 4.1.** CP-ABE 方式  $\Pi_{\text{CE}}$  が AP-IND-CCA 安全であれば, 図 4.1 の CP-ABKEM 方式  $\Pi_{\text{CK}}$  は AP-IND-CCA 安全である.

**定理 4.2.** CP-ABE 方式  $\Pi_{\text{CE}}$  が復号者アクセス構造開示性を満たすならば, 図 4.1 の CP-ABKEM 方式  $\Pi_{\text{CK}}$  は復号者アクセス構造開示性を満たす.

**定理 4.1 の証明**  $\text{Game}_0, \text{Game}_1$  の安全性ゲームを以下のように定義する.

**Game<sub>0</sub>:** 攻撃者と挑戦者が行う,  $\Pi_{\text{CK}}$  に関する AP-IND-CCA 安全性ゲーム.

**Game<sub>1</sub>:** Game<sub>0</sub> に以下の変更を加えたゲームとする.

- **Challenge フェーズ**において,  $\mathcal{CH}$  はチャレンジビット  $b$  の決定をチャレンジ鍵暗号文  $C_K^*$  の計算の前に行う. 更に, 鍵暗号文  $C_K^*$  は鍵  $K_1$  ではなく, 鍵  $K_b$  を暗号化して生成する. また,  $\mathcal{A}$  に対して, チャレンジ鍵として, チャレンジビット  $b$  によらず ( $b = 0$  であっても  $b = 1$  であっても)  $K_1$  を返す. 具体的には,  $\mathcal{CH}$  は  $\mathcal{A}$  から  $A_d^*$  を受け取り,  $K_0, K_1 \xleftarrow{\text{U}} \mathcal{K}, b \xleftarrow{\text{U}} \{0, 1\}, \text{CE.Enc}(\text{PK}, K_b, A_d) \rightarrow C_K^*$  を実行し, 最終的に  $(K_1, C_K^*)$  を  $\mathcal{A}$  へ返す.

Game <sub>$i$</sub>  の Guess フェーズで攻撃者  $\mathcal{A}$  が正しい推測ビット  $b' = b$  を出力する事象を  $W_i$  と表記する. Game<sub>0</sub> における攻撃者  $\mathcal{A}$  の利得の定義式より,

$$\text{Adv}_{\Pi_{\text{CK}}, \mathcal{A}}^{\text{AP-IND-CCA}} = |\Pr[W_0] - \frac{1}{2}| \leq |\Pr[W_0] - \Pr[W_1]| + |\Pr[W_1] - \frac{1}{2}| \quad (4.1)$$

不等式 (4.1) と以下で証明する補題 4.1.1 及び補題 4.1.2 より,  $\Pi_{\text{CE}}$  が AP-IND-CCA 安全であるならば, いかなる PPTA 攻撃者  $\mathcal{A}$  に対しても,  $\text{Adv}_{\Pi_{\text{CK}}, \mathcal{A}}^{\text{AP-IND-CCA}}$  は無視できるほど小さい. ゆえに, 定理 4.1 が成立する.  $\square$

**補題 4.1.1.** いかなる PPTA 攻撃者  $\mathcal{A}$  に対しても,  $|\Pr[W_0] - \Pr[W_1]|$  は無視できるほど小さい値になる.

**補題 4.1.2.** CP-ABE 方式  $\Pi_{\text{CE}}$  が AP-IND-CCA 安全ならば, いかなる PPTA 攻撃者  $\mathcal{A}$  に対しても,  $|\Pr[W_1] - \frac{1}{2}|$  は無視できるほど小さい値になる.

以降では補題 4.1.1 と補題 4.1.2 の証明を行う.

**補題 4.1.1 の証明**  $\text{Game}_0$  における PPTA 攻撃者を  $\mathcal{A}$  とし, 挑戦者を  $\mathcal{CH}$  とする.  $\text{Game}_0$  において  $\mathcal{A}$  と  $\mathcal{CH}$  は以下の通りに動作する.

**Setup Phase:**  $\mathcal{CH}$  は  $\text{CE.Setup}(1^k, \mathcal{U}) \rightarrow (\text{PK}, \text{MK})$  を実行し, システム公開鍵  $\text{PK}$  を  $\mathcal{A}$  に渡す.

**Query Phase 1:**  $\mathcal{A}$  は以下の各オラクルに対して, クエリを任意回数発行できる.

**秘密鍵生成:**  $\mathcal{A}$  は属性集合  $S$  を  $\mathcal{CH}$  へ送る.  $\mathcal{CH}$  は  $\text{CE.KeyGen}(\text{PK}, \text{MK}, S) \rightarrow \text{SK}$  を計算し  $\text{SK}$  を  $\mathcal{A}$  へ送る.

**鍵復号:**  $\mathcal{A}$  は鍵暗号文  $C_K$ , 属性集合  $S$  を  $\mathcal{CH}$  へ送る.  $\mathcal{CH}$  は  $\text{CE.KeyGen}(\text{PK}, \text{MK}, S) \rightarrow \text{SK}$  で秘密鍵を生成し,  $\text{CE.Dec}(\text{PK}, C_K, \text{SK}) \rightarrow K / \perp$  を実行し最後の出力結果を  $\mathcal{A}$  へ送る.

**Challenge Phase:**  $\mathcal{A}$  はターゲット復号者アクセス構造  $A_d^*$  を  $\mathcal{CH}$  へ送る.  $\mathcal{CH}$  は  $K_0, K_1 \xleftarrow{\mathcal{U}} \mathcal{K}$ ,  $\text{CE.Enc}(\text{PK}, K_1, A_d^*) \rightarrow C_K^*$ ,  $b \xleftarrow{\mathcal{U}} \{0, 1\}$  を実行し,  $(K_b, C_K^*)$  を  $\mathcal{A}$  に渡す.

**Query Phase 2:**  $\mathcal{A}$  は以下の各オラクルに対して, クエリを任意回数発行できる.

**秘密鍵生成:** Query Phase 1 と同じ.

**鍵復号:** Query Phase 1 と同じ.

**Guess Phase:**  $\mathcal{A}$  はチャレンジビット  $b$  に対する推測として,  $b' \in \{0, 1\}$  を出力.

同様に,  $\text{Game}_1$  における PPTA 攻撃者を  $\mathcal{A}$  とし, 挑戦者を  $\mathcal{CH}$  とする.  $\text{Game}_1$  において  $\mathcal{A}$  と  $\mathcal{CH}$  は以下の通りに動作する.

**Setup Phase:**  $\mathcal{CH}$  は  $\text{CE.Setup}(1^k, \mathcal{U}) \rightarrow (\text{PK}, \text{MK})$  を実行し, システム公開鍵  $\text{PK}$  を  $\mathcal{A}$  に渡す.

**Query Phase 1:**  $\mathcal{A}$  は以下の各オラクルに対して, クエリを任意回数発行できる.

**秘密鍵生成:**  $\mathcal{A}$  は属性集合  $S$  を  $\mathcal{CH}$  へ送る.  $\mathcal{CH}$  は  $\text{CE.KeyGen}(\text{PK}, \text{MK}, S) \rightarrow \text{SK}$  を計算し  $\text{SK}$  を  $\mathcal{A}$  へ送る.

**鍵復号:**  $\mathcal{A}$  は鍵暗号文  $C_K$ , 属性集合  $S$  を  $\mathcal{CH}$  へ送る.  $\mathcal{CH}$  は  $\text{CE.KeyGen}(\text{PK}, \text{MK}, S) \rightarrow \text{SK}$  で秘密鍵を生成し,  $\text{CE.Dec}(\text{PK}, C_K, \text{SK}) \rightarrow K / \perp$  を実行し最後の出力結果を  $\mathcal{A}$  へ送る.

**Challenge Phase:**  $\mathcal{A}$  はターゲット復号者アクセス構造  $A_d^*$  を  $CH$  へ送る.  $CH$  は  $K_0, K_1 \xleftarrow{U} \mathcal{K}, b \xleftarrow{U} \{0, 1\}, \text{CE.Enc}(\text{PK}, K_b, A_d^*) \rightarrow C_K^*$  を実行し,  $(K_1, C_K^*)$  を  $\mathcal{A}$  に渡す.

**Query Phase 2:**  $\mathcal{A}$  は以下の各オラクルに対して, クエリを任意回数発行できる.

**秘密鍵生成:** Query Phase 1 と同じ.

**鍵復号:** Query Phase 1 と同じ.

**Guess Phase:**  $\mathcal{A}$  はチャレンジビット  $b$  に対する推測として,  $b'$  を出力.

$\text{Game}_0$  における 5 つの変数  $K_0, K_1, C_K^*, b, b'$  の表記をそれぞれ  $K_{0,G_0}, K_{1,G_0}, C_{K,G_0}^*, b_{G_0}, b'_{G_0}$  に変更する. 同様に  $\text{Game}_1$  における 5 つの変数  $K_0, K_1, C_K^*, b, b'$  の表記をそれぞれ  $K_{0,G_1}, K_{1,G_1}, C_{K,G_1}^*, b_{G_1}, b'_{G_1}$  に変更する. 事象  $W_0$  の定義より,

$$\begin{aligned} \Pr[W_0] &= \Pr[b_{G_0} = b'_{G_0}] \\ &= \Pr[b'_{G_0} = 0 \wedge b_{G_0} = 0] + \Pr[b'_{G_0} = 1 \wedge b_{G_0} = 1] \\ &= \Pr[b_{G_0} = 0] \Pr[b'_{G_0} = 0 | b_{G_0} = 0] + \Pr[b_{G_0} = 1] \Pr[b'_{G_0} = 1 | b_{G_0} = 1] \\ &= \frac{1}{2} (\Pr[b'_{G_0} = 0 | b_{G_0} = 0] + \Pr[b'_{G_0} = 1 | b_{G_0} = 1]) \end{aligned} \quad (4.2)$$

同様に事象  $W_1$  の定義より

$$\Pr[W_1] = \Pr[b_{G_1} = b'_{G_1}] = \frac{1}{2} (\Pr[b'_{G_1} = 0 | b_{G_1} = 0] + \Pr[b'_{G_1} = 1 | b_{G_1} = 1]) \quad (4.3)$$

式(4.2), (4.3) より,

$$\begin{aligned} |\Pr[W_0] - \Pr[W_1]| &= \frac{1}{2} |\Pr[b'_{G_0} = 0 | b_{G_0} = 0] + \Pr[b'_{G_0} = 1 | b_{G_0} = 1] \\ &\quad - \Pr[b'_{G_1} = 0 | b_{G_1} = 0] - \Pr[b'_{G_1} = 1 | b_{G_1} = 1]| \\ &\leq \frac{1}{2} (|\Pr[b'_{G_0} = 0 | b_{G_0} = 0] - \Pr[b'_{G_1} = 0 | b_{G_1} = 0]| \\ &\quad + |\Pr[b'_{G_0} = 1 | b_{G_0} = 1] - \Pr[b'_{G_1} = 1 | b_{G_1} = 1]|) \end{aligned} \quad (4.4)$$

式(4.4) と, 以下に示す補題 4.1.1.1 と補題 4.1.1.2 により, いかなる PPTA 攻撃者に対しても,  $|\Pr[W_0] - \Pr[W_1]|$  は無視できるほど小さい値になる. ゆえに, 補題 4.1.1 が成立する.  $\square$

**補題 4.1.1.1.** いかなる PPTA 攻撃者に対しても,  $|\Pr[b'_{G_0} = 1 | b_{G_0} = 1] - \Pr[b'_{G_1} = 1 | b_{G_1} = 1]|$  は無視できるほど小さい値になる.

**補題 4.1.1.2.** いかなる PPTA 攻撃者に対しても,  $|\Pr[b'_{G_0} = 0 | b_{G_0} = 0] - \Pr[b'_{G_1} = 0 | b_{G_1} = 0]|$  は無視できるほど小さい値になる.

以降では, 補題 4.1.1.1 と補題 4.1.1.2 の証明を示す.

**補題 4.1.1.1 の証明** いかなる PPTA  $\mathcal{A}$  から見ても、チャレンジビットが  $b_{G_0} = 1$  に設定された  $\text{Game}_0$  の  $\mathcal{CH}$  とのやり取りと、チャレンジビットが  $b_{G_1} = 1$  に設定された  $\text{Game}_1$  の  $\mathcal{CH}$  とのやり取りは、全く同じに見えるため、当該補題が成立するという論理で証明する。

まず、 $b_{G_0} = 1$  の  $\text{Game}_0$  と  $b_{G_1} = 1$  の  $\text{Game}_1$  は、Challenge Phase 以外においては、 $\mathcal{CH}$  の動作は全く同じであるので、 $\mathcal{A}$  から見れば、 $b_{G_0} = 1$  の  $\text{Game}_0$  での Challenge Phase 以外での  $\mathcal{CH}$  とのやり取りと、 $b_{G_1} = 1$  の  $\text{Game}_1$  での Challenge Phase 以外での  $\mathcal{CH}$  とのやり取りは全く同じに見える。

次に、Challenge Phase でのやり取りに関しては、 $K_{1,G_0}$  と  $K_{1,G_1}$  はどちらも同一の鍵空間  $\mathcal{K}$  からのランダム抽出であるから、両者の分布は同じである。そして、 $K_{1,G_0}$  と  $K_{1,G_1}$  をアルゴリズム  $\text{CE.Enc}(\text{PK}, \cdot, A_d^*)$  へ入力した場合の出力結果が、それぞれ  $C_{K,G_0}^*$  と  $C_{K,G_1}^*$  であるので、 $C_{K,G_0}^*$  と  $C_{K,G_1}^*$  の分布は同じである。つまり、 $\mathcal{A}$  から見て  $(K_{1,G_0}, C_{K,G_0}^*)$  と  $(K_{1,G_1}, C_{K,G_1}^*)$  の分布は同じである。

従って、いかなる PPTA  $\mathcal{A}$  から見ても、 $b_{G_0} = 1$  の  $\text{Game}_0$  における  $\mathcal{CH}$  とのやり取りと、 $b_{G_1} = 1$  の  $\text{Game}_1$  における  $\mathcal{CH}$  とのやり取りは、同じに見える。ゆえに、補題 4.1.1.1 が成立する。  $\square$

**補題 4.1.1.2 の証明** いかなる PPTA  $\mathcal{A}$  から見ても、チャレンジビットが  $b_{G_0} = 0$  に設定された  $\text{Game}_0$  での  $\mathcal{CH}$  とのやり取りと、チャレンジビットが  $b_{G_1} = 0$  に設定された  $\text{Game}_1$  での  $\mathcal{CH}$  とのやり取りは全く同じに見えるため、当該補題が成立するという論理で証明する。

まず、 $b_{G_0} = 0$  の  $\text{Game}_0$  と  $b_{G_1} = 0$  の  $\text{Game}_1$  は、Challenge Phase 以外においては、 $\mathcal{CH}$  の動作は全く同じであるので、 $\mathcal{A}$  から見れば、 $b_{G_0} = 0$  の  $\text{Game}_0$  での Challenge Phase 以外での  $\mathcal{CH}$  とのやり取りと、 $b_{G_1} = 0$  の  $\text{Game}_1$  での Challenge Phase 以外での  $\mathcal{CH}$  とのやり取りは全く同じに見える。

次に、Challenge Phase でのやり取りに関しては、まず、 $K_{0,G_0}$  と  $K_{1,G_1}$  はどちらも同一の鍵空間  $\mathcal{K}$  からのランダム抽出であるから、 $K_{0,G_0}$  と  $K_{1,G_1}$  の分布は同じである。また、 $K_{1,G_0}$  と  $K_{0,G_1}$  はどちらも同一の鍵空間  $\mathcal{K}$  からのランダム抽出であるから、 $K_{1,G_0}$  と  $K_{0,G_1}$  の分布は同じである。そして、 $K_{1,G_0}$  と  $K_{0,G_1}$  をアルゴリズム  $\text{CE.Enc}(\text{PK}, \cdot, A_d^*)$  へ入力した場合の出力結果が、それぞれ  $C_{K,G_0}^*$  と  $C_{K,G_1}^*$  であるので、 $C_{K,G_0}^*$  と  $C_{K,G_1}^*$  の分布は同じである。ここで、 $K_{0,G_0}$  と  $K_{1,G_0}$  は独立に選ばれていることから、 $K_{0,G_0}$  と  $C_{K,G_0}^*$  は独立の関係にある。同様に、 $K_{1,G_1}$  と  $C_{K,G_1}^*$  も独立の関係にある。従って、 $\mathcal{A}$  から見て  $(K_{0,G_0}, C_{K,G_0}^*)$  と  $(K_{1,G_1}, C_{K,G_1}^*)$  の分布は同じである。

従って、いかなる PPTA  $\mathcal{A}$  から見ても、 $b_{G_0} = 0$  の  $\text{Game}_0$  における  $\mathcal{CH}$  とのやり取りと、 $b_{G_1} = 0$  の  $\text{Game}_1$  における  $\mathcal{CH}$  とのやり取りは、同じに見える。ゆえに、補題 4.1.1.2 が成立する。  $\square$

**補題 4.1.2 の証明** PPTA 攻撃者  $\mathcal{A}$  は  $\text{Game}_1$  の安全性ゲームで無視できない利得でゲームに勝利できると仮定する。PPTA シミュレータ  $\mathcal{S}$  は、 $\mathcal{A}$  に対して  $\text{Game}_1$  を完璧にシミュレートし、 $\mathcal{A}$  の Guess Phase での最終的な出力を利用して、 $\Pi_{\text{CE}}$  に関する AP-IND-

CCA 安全性ゲームに勝利しようとする.  $\Pi_{\text{CE}}$  に関する AP-IND-CCA 安全性ゲームにおける挑戦者を  $\mathcal{CH}$  と表記する.  $\mathcal{A}$ ,  $\mathcal{S}$ ,  $\mathcal{CH}$  の動作は以下の通りである.

**Setup Phase:**  $\mathcal{S}$  は,  $\mathcal{CH}$  よりシステム公開鍵  $\text{PK}$  を受け取り,  $\text{PK}$  を  $\mathcal{A}$  へ送る.

**Query Phase 1:**  $\mathcal{A}$  から  $\mathcal{S}$  への各オラクルクエリに対する  $\mathcal{S}$  の動作は以下の通りである.

**秘密鍵生成:**  $\mathcal{A}$  は属性集合  $S$  をクエリする.  $\mathcal{S}$  は  $\Pi_{\text{CE}}$  に関する AP-IND-CCA 安全性ゲームの Query Phase 1 の秘密鍵生成オラクルへのクエリとして  $S$  を発行し,  $\text{SK}$  を受け取る.  $\mathcal{S}$  は  $\text{SK}$  を  $\mathcal{A}$  へ送る.

**鍵復号:**  $\mathcal{A}$  は鍵暗号文  $C_K$ , 属性集合  $S$  をクエリする.  $\mathcal{S}$  は Query Phase 1 の復号オラクルクエリとして  $(C_K, S)$  を発行し,  $m / \perp =: \alpha$  を受け取る.  $\mathcal{S}$  は  $\alpha$  を  $\mathcal{A}$  へ送る.

**Challenge Phase:**  $\mathcal{A}$  はターゲット復号者アクセス構造  $A_d^*$  を出力する.  $\mathcal{S}$  は  $K_0, K_1 \xleftarrow{\mathcal{U}} \mathcal{K}$  を実行し, ターゲット平文, ターゲット復号者アクセス構造として,  $(K_0, K_1, A_d^*)$  を  $\mathcal{CH}$  へ送り, チャレンジ暗号文  $C_K^*$  を受け取る.  $\mathcal{S}$  は,  $(K_1, C_K^*)$  を  $\mathcal{A}$  へ送る.

**Query Phase 2:**  $\mathcal{A}$  から  $\mathcal{S}$  への各オラクルクエリに対する  $\mathcal{S}$  の動作は以下の通りである.

**秘密鍵生成:** Query Phase 1 と同じ.

**鍵復号:** Query Phase 1 と同じ.

**Guess Phase:**  $\mathcal{A}$  チャレンジビット  $b$  に対する推測としてビット  $b'$  を出力し,  $\mathcal{S}$  は  $b'$  を  $\mathcal{CH}$  に対して出力する.

$\Pi_{\text{CE}}$  に関する AP-IND-CCA 安全性ゲームのチャレンジビットを  $b$  とすると,  $\mathcal{S}$  が  $\mathcal{A}$  に対して, チャレンジビットが  $b$  である  $\text{Game}_1$  を完璧にシミュレートできていることはほぼ自明であるため詳細な説明は割愛する.

$\mathcal{S}$  が  $\Pi_{\text{CK}}$  の AP-IND-CCA 安全性ゲームにおいてルール上禁止されたオラクルクエリを一度も行わないことを示す.  $\mathcal{S}$  が  $\Pi_{\text{CK}}$  の AP-IND-CCA ゲームにおいてルール上禁止されたオラクルクエリを発行する可能性のあるタイミングは以下に示す三つである (丸括弧“( )”内は  $\mathcal{S}$  がそのオラクルクエリを発行するきっかけとなる  $\mathcal{A}$  から  $\mathcal{S}$  へのクエリが行われるオラクルを表す). (1)Query Phase 1 の秘密鍵生成 (Query Phase 1 の秘密鍵生成), (2)Query Phase 2 の秘密鍵生成 (Query Phase 2 の秘密鍵生成), (3)Query Phase 2 の復号 (Query Phase 2 の鍵復号). (1), (2), (3) いずれに関しても,  $\mathcal{S}$  にとって

の禁止クエリは、 $\mathcal{A}$  にとっての禁止クエリでもある。具体的には、 $\mathcal{S}$  が秘密鍵生成オラクルで  $S_r \in \mathbb{A}_d^*$  を満たす  $S_r$  をクエリすることは禁止だが、 $\mathcal{A}$  が秘密鍵生成オラクルで同じ条件を満たす  $S_r$  をクエリすることはそもそも禁止である。また、 $\mathcal{S}$  が Query Phase 2 の復号オラクルで  $C_K = K^*$  かつ  $S_r \in \mathbb{A}_d^*$  を満たす  $(C_K, S_r)$  をクエリすることは禁止だが、 $\mathcal{A}$  が Query Phase 2 の鍵復号オラクルで同じ条件を満たす  $(C_K, S_r)$  をクエリすることはそもそも禁止である。よって、本証明内では  $\mathcal{A}$  が禁止クエリを発行することは一度もないと仮定できることから、 $\mathcal{S}$  が禁止クエリを発行することは一度もない。従って、以下の等式が成立する。

$$\text{Adv}_{\Pi_{\text{CE}}, \mathcal{S}}^{\text{AP-IND-CCA}} = \text{Adv}_{\Pi_{\text{CK}}, \mathcal{A}}^{\text{Game}_1} = |\Pr[W_1] - \frac{1}{2}| \quad (4.5)$$

$|\Pr[W_1] - \frac{1}{2}|$  が無視できなくなるような PPTA  $\mathcal{A}$  が存在すると仮定すると、式 (4.5) より、 $\text{Adv}_{\Pi_{\text{CE}}, \mathcal{S}}^{\text{AP-IND-CCA}}$  は無視できなくなり、これは  $\Pi_{\text{CE}}$  が AP-IND-CCA 安全であるという事実に矛盾するので、その仮定は誤りである。従って、 $\Pi_{\text{CE}}$  が AP-IND-CCA 安全ならば、いかなる PPTA 攻撃者  $\mathcal{A}$  に対しても、 $|\Pr[W_1] - \frac{1}{2}|$  は無視できるほど小さい値になる。ゆえに、補題 4.1.2 が成立する。□

**定理 4.2 の証明** CP-ABE 方式  $\Pi_{\text{CE}}$  に関して、復号者アクセス構造開示性の定義を満たすアルゴリズムを、 $\text{Disclose}_{\text{CE}}$  と表記する。

全ての  $k$ 、全ての  $\mathcal{U}$ 、全ての  $(\text{PK}, \text{MK}) \leftarrow \text{CE.Setup}(1^k, \mathcal{U})$ 、全ての  $m$ 、全ての  $S \in (2^{\mathcal{U}} - \{\emptyset\})$ 、全ての  $\mathbb{A}_d$ 、全ての  $K \stackrel{\mathcal{U}}{\leftarrow} \mathcal{K}$ 、全ての  $C_K \leftarrow \text{CE.Enc}(\text{PK}, K, \mathbb{A}_d)$  に対して、アルゴリズム  $\text{Disclose}_{\text{CE}}$  をサブルーチンとして利用し、 $C_K$  を入力変数とするアルゴリズム  $\text{Disclose}_{\text{CK}} \Gamma$  図 6.2 のように定義する。

$\text{Disclose}_{\text{CK}}(\text{PK}, C_K) :$   
Return  $\text{Disclose}_{\text{CE}}(\text{PK}, C_K)$ .

図 4.2: 図 4.1 の CP-ABKEM 方式  $\Pi_{\text{CK}}$  の復号者アクセス構造開示性アルゴリズム  $\text{Disclose}_{\text{CK}}$

$\text{Disclose}_{\text{CK}}$  の定義より、次の等式が成立する。

$$\text{Disclose}_{\text{CK}}(\text{PK}, C_K) = \text{Disclose}_{\text{CE}}(\text{PK}, C_K) \quad (4.6)$$

$\text{Disclose}_{\text{CE}}$  は  $\Pi_{\text{CE}}$  の復号者アクセス構造開示性アルゴリズムだから、CP-ABE 方式の復号者アクセス構造開示性の定義より、次の等式が成立する。

$$\Pr[\text{Disclose}_{\text{CE}}(\text{PK}, C_K) = \mathbb{A}_d] = 1 \quad (4.7)$$

式 (4.6), (4.7) より、次の等式が成立する。

$$\Pr[\text{Disclose}_{\text{CK}}(\text{PK}, C_K) = \mathbb{A}_d] = 1 \quad (4.8)$$

式 (4.8) と CP-ABKEM 方式の復号者アクセス構造開示性の定義より，アルゴリズム  $\text{Disclose}_{\text{CK}}$  は復号者アクセス構造開示性アルゴリズムとしての条件を満たしている．よって，図 4.1 の CP-ABKEM 方式  $\Pi_{\text{CK}}$  は復号者アクセス構造開示性を満たす．ゆえに，定理 4.2 は成立する．  $\square$

# Chapter 5 暗号文ポリシー型属性ベース Signcryptionの一般的構成

## 5.1 本章の概要

本章に示す成果は、暗号文ポリシー型属性ベース暗号 (CP-ABKEM), 署名ポリシー型属性ベース署名 (SP-ABS), データカプセル化メカニズム (DEM) を構成要素とする, 暗号文ポリシー型属性ベース Signcryption (CP-ABSC) の一般的構成法とその安全性証明である. 具体的には, **AP-IND-CCA** 安全かつ復号者アクセス構造開示的な CP-ABKEM, **AP-sEUF-CMA** 安全かつ完全匿名かつ署名者アクセス構造衝突困難な SP-ABS, **IND-CCA** 安全かつ一対一対応な DEM から, **AP-IND-CCA** 安全かつ **AP-sEUF-CMA** 安全かつ完全匿名な CP-ABSC を一般的に構成できることを証明した.

なお, 本成果 (暗号文ポリシー型属性ベース Signcryption の一般的構成) の意義については, 8.1.2 項で説明する.

また, 本章の構成については以下の通りである. 5.2 節で一般的構成法の説明を行う. 5.3 節で安全性証明の詳述を行う.

## 5.2 提案する一般的構成法

構成要素として, CP-ABKEM 方式  $\Pi_{CK}:(CK.Setup,CK.KeyGen,CK.Encap,CK.Decap)$ , DEM 方式  $\Pi_D:(D.Encap,D.Decap)$ , SP-ABS 方式  $\Pi_{SS}:(SS.Setup,SS.KeyGen,SS.Sig,SS.Ver)$  の三つの要素技術を用いた CP-ABSC の一般的構成法  $\Pi_{CS}:(CS.Setup,CS.KeyGen_S,CS.KeyGen_R,CS.SC,CS.USC)$  を図 5.1 に示す. 当該構成法は, Chiba ら [31] が提案した, Signcryption にとっての最強の安全性とみなされている「多人数モデルにおける内部攻撃者に対する IND-CCA 安全性と sEUF-CMA 安全性」を達成可能であることを示した Signcryption の一般的構成法の“CP-ABSC 版”への拡張とみなすことができる.

以下, 図 5.1 の構成法を採用した理由について, 説明する.

まず, 公開鍵暗号とデジタル署名の両機能を実現する Signcryption に関しては, 代表的かつ自明な構成法として, “Encrypt-then-Sign” と, “Sign-then-Encrypt” と呼ばれる構成法が存在する. Encrypt-then-Sign は, 最初に平文を暗号化し, その暗号文に対する署名作成を行い, 暗号文と署名を連結したものをサインクリプトテキストとするような, 極めて単純な構成法である. 対して, Sign-then-Encrypt は, 最初に平文に対する署名作成を行い, その後で作成した署名を暗号化し, その暗号文をサインクリプトテキストとするような, 極めて単純な構成法である. Encrypt-then-Sign も, Sign-then-Encrypt も, Signcryption にとっての最強の安全性を達成することは不可能であることが知られ

$\text{CS.Setup}(1^k, \mathcal{U}_s, \mathcal{U}_r) :$ $(\text{PK}_{ss}, \text{MK}_{ss}) \leftarrow \text{SS.Setup}(1^k, \mathcal{U}_s);$ $(\text{PK}_{ck}, \text{MK}_{ck}) \leftarrow \text{CK.Setup}(1^k, \mathcal{U}_r);$ $\text{Return } (\text{PK}, \text{MK}) := ((\text{PK}_{ss}, \text{PK}_{ck}), (\text{MK}_{ss}, \text{MK}_{ck})).$
$\text{CS.KeyGen}_S(\text{PK}, \text{MK}, S_s) :$ $\text{Return } \text{SK}_s \leftarrow \text{SS.KeyGen}(\text{PK}_{ss}, \text{MK}_{ss}, S_s).$
$\text{CS.KeyGen}_R(\text{PK}, \text{MK}, S_r) :$ $\text{Return } \text{SK}_r \leftarrow \text{CK.KeyGen}(\text{PK}_{ck}, \text{MK}_{ck}, S_r).$
$\text{CS.SC}(\text{PK}, m, \text{SK}_s, \mathbb{A}_s, \mathbb{A}_d) :$ $(K, C_K) \leftarrow \text{CK.Encap}(\text{PK}_{ck}, \mathbb{A}_d);$ $\sigma \leftarrow \text{SS.Sig}(\text{PK}_{ss}, m \  C_K, \text{SK}_s, \mathbb{A}_s);$ $C_D \leftarrow \text{D.Encap}(K, m \  \sigma);$ $\text{Return } C := (C_K, C_D).$
$\text{CS.USC}(\text{PK}, C, \text{SK}_r, \mathbb{A}_s) :$ $\text{Parse } C \text{ as } (C_K, C_D).$ $\alpha := \text{CK.Decap}(\text{PK}_{ck}, C_K, \text{SK}_r);$ $\text{If } \alpha = \perp, \text{ then return } \perp. \text{ Else } K := \alpha.$ $\beta := \text{D.Decap}(K, C_D);$ $\text{If } \beta = \perp, \text{ then return } \perp. \text{ Else } m \  \sigma := \beta.$ $\gamma := \text{SS.Ver}(\text{PK}_{ss}, \sigma, m \  C_K, \mathbb{A}_s);$ $\text{If } \gamma = 0, \text{ then return } \perp. \text{ Else return } m.$

図 5.1: 提案する CP-ABSC 方式の一般的構成法  $\Pi_{\text{CS}}$ 

ている．具体的には，Encrypt-then-Sign の場合，多人数モデルにおける内部送信者に対する IND-CCA 安全性ゲームにおいて，攻撃者が実行することで確実に (100%) ゲームに勝利できる攻撃法が存在するために，Encrypt-then-Sign は当該安全性を達成不可能であることが知られている．また，Sign-then-Encrypt の場合，多人数モデルにおける内部受信者に対する sEUF-CMA 安全性ゲームにおいて，攻撃者が実行することで確実に (100%) ゲームに勝利できる攻撃法が存在するために，Sign-then-Encrypt は当該安全性を達成不可能であることが知られている．

また，暗号文ポリシー型属性ベース Signcryption(CP-ABSC) に関しても，Signcryption と同様に，暗号文ポリシー型属性ベース暗号 (CP-ABE) と署名ポリシー型属性ベース署名 (SP-ABS) を用いて，最も単純な構成に基づく “Encrypt-then-Sign” 式の構成法  $\Pi_{\text{CS}(\text{EtS})} : (\text{CS}(\text{EtS}).\text{Setup}, \text{CS}(\text{EtS}).\text{KeyGen}_S, \text{CS}(\text{EtS}).\text{KeyGen}_R, \text{CS}(\text{EtS}).\text{SC}, \text{CS}(\text{EtS}).\text{USC})$  と，同じく最も単純な構成に基づく “Sign-then-Encrypt” 式の構成法  $\Pi_{\text{CS}(\text{StE})} : (\text{CS}(\text{StE}).\text{Setup}, \text{CS}(\text{StE}).\text{KeyGen}_S, \text{CS}(\text{StE}).\text{KeyGen}_R, \text{CS}(\text{StE}).\text{SC}, \text{CS}(\text{StE}).\text{USC})$  を，それぞれ図 5.2 と図 5.3 の通りに定義することができる．しかし，2.8.1 項で説明したように，CP-ABSC の AP-IND-CCA 安全性が実際には内部者安全性であることによって，Encrypt-then-Sign 式構成法  $\Pi_{\text{CS}(\text{EtS})}$  の場合，AP-IND-CCA 安全性

ゲームにおいて、攻撃者が、“Signcryption の Encrypt-then-Sign 式構成法に関する多人数モデルにおける内部攻撃者に対する IND-CCA 安全性ゲームにおいて攻撃者が実行することで確実にゲームに勝利することができる攻撃法”を“CP-ABSC 版へ拡張した攻撃法”を実行することで、AP-IND-CCA ゲームに確実に勝利できる。従って、Encrypt-then-Sign 式構成法  $\Pi_{CS(ETs)}$  は、AP-IND-CCA 安全性を達成不可能である。同様に、2.8.3 項で説明したように、CP-ABSC の AP-sEUF-CMA 安全性が実際には内部者安全性であることによって、Sign-then-Encrypt 式構成法  $\Pi_{CS(STE)}$  の場合、AP-sEUF-CMA 安全性ゲームにおいて、攻撃者が、“Signcryption の Sign-then-Encrypt 式構成法に関する多人数モデルにおける内部受信者に対する sEUF-CMA 安全性ゲームにおいて攻撃者が実行することで確実にゲームに勝利することができる攻撃法”を“CP-ABSC 版へ拡張した攻撃法”を、実行することで、AP-sEUF-CMA ゲームに確実に勝利できる。従って、Sign-then-Encrypt 式構成法  $\Pi_{CS(STE)}$  は、AP-sEUF-CMA 安全性を達成不可能である。

以下、構成法  $\Pi_{CS(ETs)}$  が AP-IND-CCA 安全性を達成する上でその達成を阻む攻撃法と、構成法  $\Pi_{CS(STE)}$  が AP-sEUF-CMA 安全性を達成する上でその達成を阻む攻撃法のそれぞれを簡単に説明する。

第一に、Encrypt-then-Sign 式の構成法  $\Pi_{CS(ETs)}$  に関する AP-IND-CCA 安全性ゲームにおいて、攻撃者は図 5.4 に示すような攻撃法を実行することによって、ゲームに 100% 勝利することができるため、Encrypt-then-Sign 式の構成法  $\Pi_{CS(ETs)}$  は AP-IND-CCA 安全性を達成できない。

以下、図 5.4 の攻撃手順について、説明する。この攻撃法を実行することによって、攻撃者は構成法  $\Pi_{CS(ETs)}$  に関する AP-IND-CCA ゲームに 100% 勝利することができる。

1. 攻撃者は、チャレンジフェーズで挑戦者に適当なクエリ  $(m_0, m_1, A_s^*, A_d^*, S_s^*)$  を送り、挑戦者は当該図に示すような適切な手順でチャレンジサインクリプテキスト  $C^* := (C_1^*, \sigma^*)$  を作成し、攻撃者に  $C^*$  を送る。
2. 攻撃者は、送信者用秘密鍵生成オラクルに、 $S'_s \in A_s^*$  を満たす属性集合  $S'_s$  をクエリし、対応する秘密鍵  $SK'_s$  を受け取る。本稿の CP-ABSC の AP-IND-CCA の安全性定義からも明らかのように、攻撃者がこのようなクエリを行うことは許されている。
3. 攻撃者は、入手した秘密鍵  $SK'_s$  を利用して、チャレンジサインクリプテキスト  $C^*$  の第一成分  $C_1^*$  に対するターゲット署名者アクセス構造  $A_s^*$  の下での正当な署名  $\sigma'$  を作成する。ここで、 $\sigma' = \sigma^*$  であれば、再度同じ手順で署名  $\sigma'$  の作成を行い、 $\sigma' \neq \sigma^*$  となるまで、これを繰り返す。その後、生成した署名  $\sigma'$  を第二成分とする新たなサインクリプテキスト  $C' := (C_1^*, \sigma')$  を作成する。
4. 攻撃者は、アンサインクリプションオラクルへ、 $C', A_s^*, S'_r (s.t. S'_r \in A_d^*)$  をクエリする。本稿の CP-ABSC の AP-IND-CCA の安全性定義からも明らか

$\text{CS(EtS).Setup}(1^k, \mathcal{U}_s, \mathcal{U}_r) :$ $(\text{PK}_{ss}, \text{MK}_{ss}) \leftarrow \text{SS.Setup}(1^k, \mathcal{U}_s);$ $(\text{PK}_{ce}, \text{MK}_{ce}) \leftarrow \text{CE.Setup}(1^k, \mathcal{U}_r);$ $\text{Return } (\text{PK}, \text{MK}) := ((\text{PK}_{ss}, \text{PK}_{ce}), (\text{MK}_{ss}, \text{MK}_{ce})).$
$\text{CS(EtS).KeyGen}_S(\text{PK}, \text{MK}, S_s) :$ $\text{Return } \text{SK}_s \leftarrow \text{SS.KeyGen}(\text{PK}_{ss}, \text{MK}_{ss}, S_s).$
$\text{CS(EtS).KeyGen}_R(\text{PK}, \text{MK}, S_r) :$ $\text{Return } \text{SK}_r \leftarrow \text{CE.KeyGen}(\text{PK}_{ce}, \text{MK}_{ce}, S_r).$
$\text{CS(EtS).SC}(\text{PK}, m, \text{SK}_s, \mathbb{A}_s, \mathbb{A}_d) :$ $C_1 \leftarrow \text{CE.Enc}(\text{PK}_{ce}, m, \mathbb{A}_d);$ $\sigma \leftarrow \text{SS.Sig}(\text{PK}_{ss}, C_1, \text{SK}_s, \mathbb{A}_s);$ $\text{Return } C := (C_1, \sigma).$
$\text{CS(EtS).USC}(\text{PK}, C, \text{SK}_r, \mathbb{A}_s) :$ $\text{Parse } C \text{ as } (C_1, \sigma).$ $\alpha := \text{CE.Dec}(\text{PK}_{ce}, C_1, \text{SK}_r);$ $\text{If } \alpha = \perp, \text{ then return } \perp. \text{ Else } m := \alpha.$ $\beta := \text{SS.Ver}(\text{PK}_{ss}, \sigma, C_1, \mathbb{A}_s);$ $\text{If } \beta = 0, \text{ then return } \perp. \text{ Else return } m.$

図 5.2: “Encrypt-then-Sign” 式の CP-ABSC の一般的構成法のうち、構成が最も単純な構成法  $\Pi_{\text{CS(EtS)}}$

$\text{CS(StE).Setup}(1^k, \mathcal{U}_s, \mathcal{U}_r) :$ $(\text{PK}_{ss}, \text{MK}_{ss}) \leftarrow \text{SS.Setup}(1^k, \mathcal{U}_s);$ $(\text{PK}_{ce}, \text{MK}_{ce}) \leftarrow \text{CE.Setup}(1^k, \mathcal{U}_r);$ $\text{Return } (\text{PK}, \text{MK}) := ((\text{PK}_{ss}, \text{PK}_{ce}), (\text{MK}_{ss}, \text{MK}_{ce})).$
$\text{CS(StE).KeyGen}_S(\text{PK}, \text{MK}, S_s) :$ $\text{Return } \text{SK}_s \leftarrow \text{SS.KeyGen}(\text{PK}_{ss}, \text{MK}_{ss}, S_s).$
$\text{CS(StE).KeyGen}_R(\text{PK}, \text{MK}, S_r) :$ $\text{Return } \text{SK}_r \leftarrow \text{CE.KeyGen}(\text{PK}_{ce}, \text{MK}_{ce}, S_r).$
$\text{CS(StE).SC}(\text{PK}, m, \text{SK}_s, \mathbb{A}_s, \mathbb{A}_d) :$ $\sigma \leftarrow \text{SS.Sig}(\text{PK}_{ss}, m, \text{SK}_s, \mathbb{A}_s);$ $C \leftarrow \text{CE.Enc}(\text{PK}_{ce}, m \parallel \sigma, \mathbb{A}_d);$ $\text{Return } C.$
$\text{CS(StE).USC}(\text{PK}, C, \text{SK}_r, \mathbb{A}_s) :$ $\alpha := \text{CE.Dec}(\text{PK}_{ce}, C, \text{SK}_r);$ $\text{If } \alpha = \perp, \text{ then return } \perp. \text{ Else } m \parallel \sigma := \alpha.$ $\beta := \text{SS.Ver}(\text{PK}_{ss}, \sigma, m, \mathbb{A}_s);$ $\text{If } \beta = 0, \text{ then return } \perp. \text{ Else return } m.$

図 5.3: “Sign-then-Encrypt” 式の CP-ABSC の一般的構成法のうち、構成が最も単純な構成法  $\Pi_{\text{CS(StE)}}$

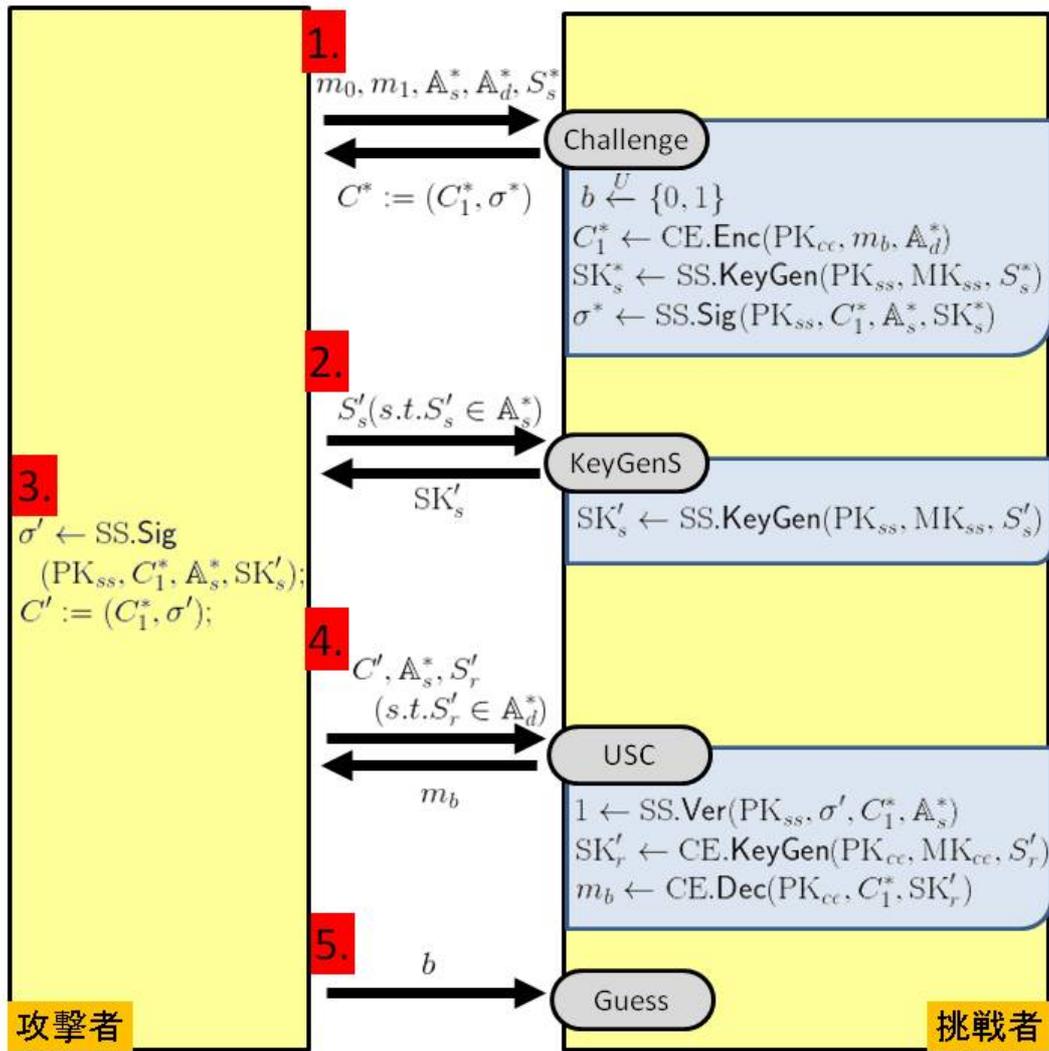


図 5.4: Encrypt-then-Sign 式の構成法  $\Pi_{\text{CS(ETS)}}$  を, AP-IND-CCA 安全性達成不可能にする攻撃手順

かなように、攻撃者がこのようなクエリを行うことは許されている。その後、挑戦者は  $\text{SS.Ver}(\text{PK}_{ss}, \sigma', C_1^*, \mathbb{A}_s^*)$  を実行する。この処理の出力結果は、 $C_1^*$  の生成過程を考えると、SP-ABS 方式  $\Pi_{\text{SS}}$  の正当性より確率 1 で “1” となることは自明である。その後、挑戦者は  $S_r'$  に対応する秘密鍵  $\text{SK}_r'$  を作成し、 $\text{SK}_r'$  を利用して、 $\text{CE.Dec}(\text{PK}_{ce}, C_1^*, \text{SK}_r')$  を実行する。この処理の出力結果は、 $S_r \in \mathbb{A}_d^*$  であることを考えると、CP-ABE 方式  $\Pi_{\text{CE}}$  の正当性より確率 1 で “ $m_b$ ” となることは自明である。最後に、挑戦者は  $m_b$  を攻撃者に送る。

5. 攻撃者は、手順 4 で  $m_b$  を手に入れ、手順 1 で自身が決定した  $m_0, m_1$  は記憶しているので、チャレンジビット  $b$  を正しく推測し、Guess フェーズで出力することができる。

第二に、Sign-then-Encrypt 式の構成法  $\Pi_{\text{CS(StE)}}$  に関する AP-sEUF-CMA 安全性ゲームにおいて、攻撃者は図 5.5 に示すような攻撃法を実行することによって、ゲームに 100% 勝利することができるため、単純な Sign-then-Encrypt 式の構成法  $\Pi_{\text{CS(StE)}}$  は AP-sEUF-CMA 安全性を達成できない。

以下、図 5.5 の攻撃手順について、説明する。この攻撃法を実行することによって、攻撃者は構成法  $\Pi_{\text{CS(StE)}}$  に関する AP-sEUF-CMA ゲームに 100% 勝利することができる。

1. 攻撃者は、サインクリプションオラクルに、適当なクエリ  $m, \mathbb{A}_s, \mathbb{A}_d, S_s$  を送る。挑戦者は、当該図に示す手順でサインクリプトテキスト  $C$  を作成し、 $C$  を攻撃者に送る。また、 $(m, C, \mathbb{A}_s, \mathbb{A}_d)$  をリスト  $\mathcal{L}_{\text{SC}}$  へ追加する。
2. 攻撃者は、 $S_r \in \mathbb{A}_d$  を満たす属性集合  $S_r$  を、受信者用秘密鍵生成オラクルへクエリする。挑戦者は、 $\text{SK}_r \leftarrow \text{CE.KeyGen}(\text{PK}_{ce}, \text{MK}_{ce}, S_r)$  を実行し、 $\text{SK}_r$  を攻撃者へ送る。本稿の CP-ABSC の AP-sEUF-CMA の安全性定義からも明らかのように、攻撃者がこのようなクエリを行うことは許されている。
3. 攻撃者は、サインクリプトテキストでありかつ暗号文でもある  $C$  に関して、 $\text{CE.Dec}(\text{PK}_{ce}, C, \text{SK}_r)$  を実行する。その実行結果は、 $\text{SK}_r$  が属性集合  $S_r$  に対応する秘密鍵であり、かつ  $S_r$  が  $S_r \in \mathbb{A}_d$  を満たすことを考慮すると、CP-ABE 方式  $\Pi_{\text{CE}}$  の正当性より確率 1 で “ $m||\sigma$ ” となる。続けて、攻撃者は  $\text{CE.Enc}(\text{PK}_{ce}, m||\sigma, \mathbb{A}_d)$  を実行し、暗号文あるいはサインクリプトテキスト  $C'$  を作成する。ここで、 $C' = C$  であれば、攻撃者は再度  $\text{CE.Enc}(\text{PK}_{ce}, m||\sigma, \mathbb{A}_d)$  を実行し、 $C'$  を作成し、これを  $C' \neq C$  となるまで繰り返す。CE.Enc アルゴリズムが一般的に確率的アルゴリズムであることから、 $C' \neq C$  を

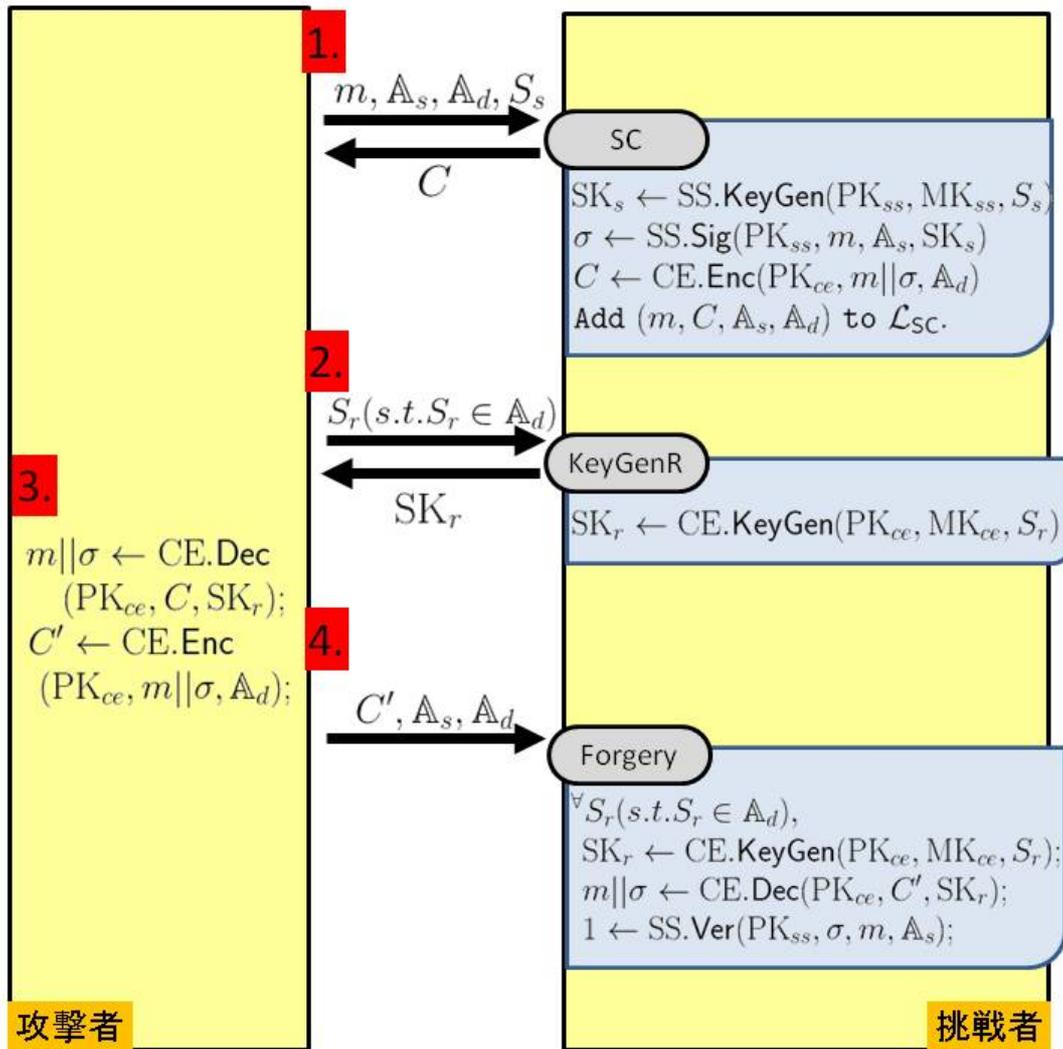


図 5.5: Sign-then-Encrypt 式の構成法  $\Pi_{CS(StE)}$  を, AP-sEUF-CMA 安全性達成不可能にする攻撃手順.

満たす  $C'$  は容易に作成できる.

4. 攻撃者は, **Forgery** フェーズで,  $C', \mathbb{A}_s, \mathbb{A}_d$  を出力する. 挑戦者は,  $S_r \in \mathbb{A}_d$  を満たすある属性集合  $S_r$  について,  $\text{SK}_r \leftarrow \text{CE.KeyGen}(\text{PK}_{ck}, \text{MK}_{ck}, S_r)$  を実行する. 続けて, 挑戦者は  $\alpha := \text{CE.Dec}(\text{PK}_{ce}, C', \text{SK}_r)$  を実行する. **CP-ABE** 方式  $\Pi_{\text{CE}}$  の正当性より, 確率 1 で,  $\alpha = m \parallel \sigma$  となる. 続けて, 挑戦者は  $\beta := \text{SS.Ver}(\text{PK}_{ss}, \sigma, m, \mathbb{A}_s)$  を実行する. **SP-ABS** 方式  $\Pi_{\text{SS}}$  の正当性より, 確率 1 で,  $\beta = 1$  となる. 以上は,  $S_r \in \mathbb{A}_d$  を満たすある属性集合  $S_r$  に対しての処理であったが,  $S'_r \in \mathbb{A}_d$  を満たす他の全ての属性集合  $S'_r$  についても, 同様に変数  $\beta$  を定義するとすれば, 必ず  $\beta = 1$  になる. さらに,  $C' \neq C$  であることから,  $(m, C', \mathbb{A}_s, \mathbb{A}_d) \notin \mathcal{L}_{\text{SC}}$  が成り立つことを考慮すると, 攻撃者はこのゲームに必ず勝利する.

図 5.1 の構成法は, **Sign-then-Encrypt** 式の **CP-ABSC** の一般的構成法の一つであるが, **Sign-then-Encrypt** 式の **CP-ABSC** の一般的構成法の中で構成が最も単純である構成法  $\Pi_{\text{CS(StE)}}$  との違いは, 構成要素の一つである **DEM** 方式  $\Pi_{\text{D}}$  に一対一対応性を仮定することにより, 構成法  $\Pi_{\text{CS(StE)}}$  に対しては有効であった **AP-sEUF-CMA** 安全性を破る攻撃法 (図 5.5) を無効化することができる点である. 以下, その詳細について説明する.

図 5.1 の構成法に関する **AP-sEUF-CMA** ゲームにおいて, 攻撃者が図 5.5 の攻撃と同様の攻撃を試みる場合を考える. 当該攻撃に関して, イメージ図を図 5.6 に示し, 攻撃手順を以下で詳述する.

1. 攻撃者は, サインクリプションオラクルに, 適当なクエリ  $m, \mathbb{A}_s, \mathbb{A}_d, S_s$  を送る. 挑戦者は, 当該図に示す手順でサインクリプトテキスト  $C := (C_K, C_D)$  を作成し,  $C$  を攻撃者に送る. また,  $(m, C = (C_K, C_D), \mathbb{A}_s, \mathbb{A}_d)$  をリスト  $\mathcal{L}_{\text{SC}}$  へ追加する.
2. 攻撃者は,  $S_r \in \mathbb{A}_d$  を満たす属性集合  $S_r$  を, 受信者用秘密鍵生成オラクルへクエリする. 挑戦者は,  $\text{SK}_r \leftarrow \text{CK.KeyGen}(\text{PK}_{ck}, \text{MK}_{ck}, S_r)$  を実行し,  $\text{SK}_r$  を攻撃者へ送る. 本稿の **CP-ABSC** の **AP-sEUF-CMA** の安全性定義からも明らかのように, 攻撃者がこのようなクエリを行うことは許されている.
3. 攻撃者は, サインクリプトテキスト  $C$  の第一成分  $C_K$  に関して,  $\text{CK.Dec}(\text{PK}_{ck}, C_D, \text{SK}_r)$  を実行する. その実行結果は,  $\text{SK}_r$  が属性集合  $S_r$  に対応する秘密鍵であり, かつ  $S_r$  が  $S_r \in \mathbb{A}_d$  を満たすことを考慮すると, **CP-ABKEM** 方式  $\Pi_{\text{CK}}$  の正当性より確率 1 で “ $m \parallel \sigma$ ” となる. 続けて, 攻撃者は, サインクリプトテキスト  $C$  の第二成分  $C_D$  に関して,

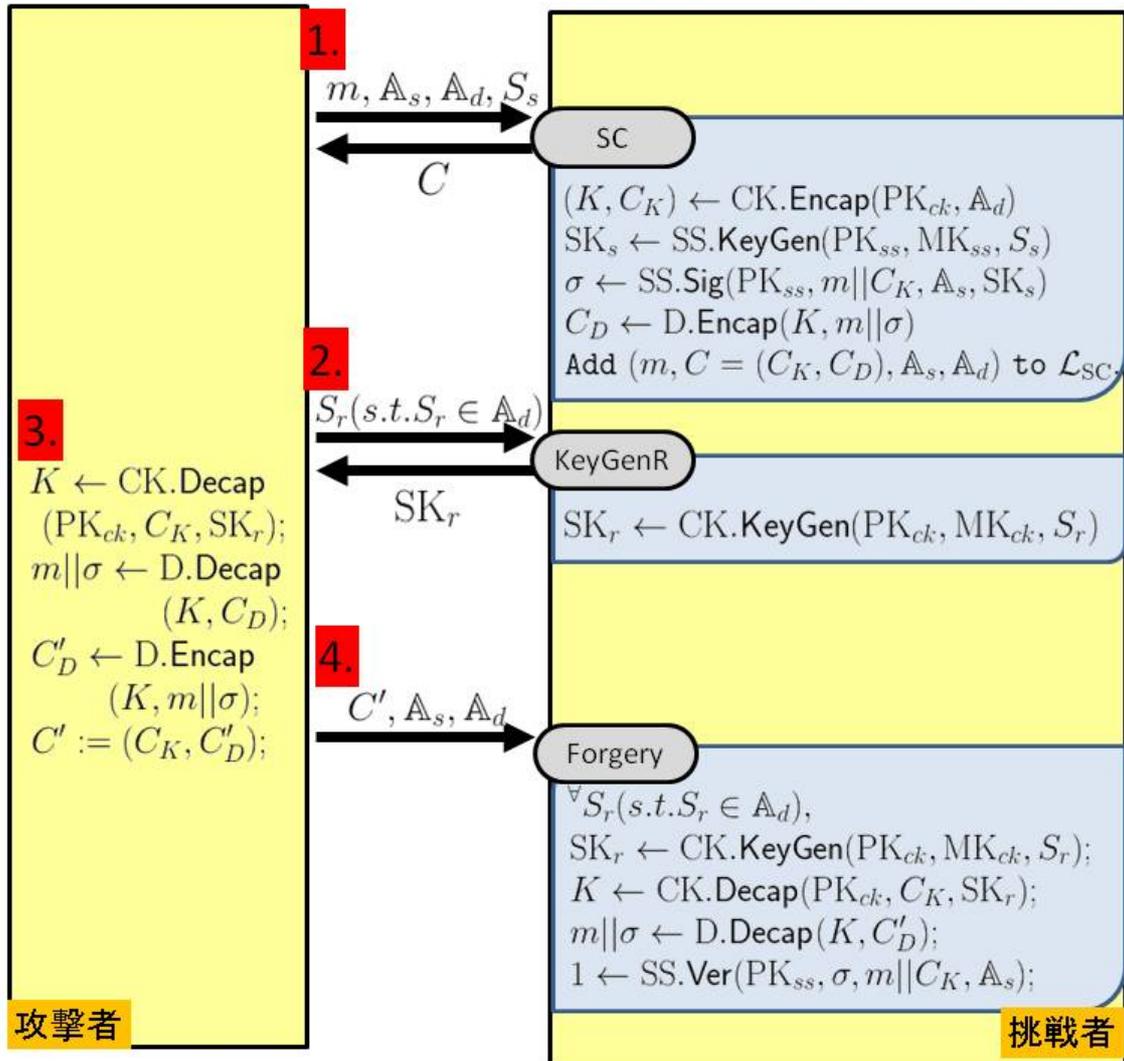


図 5.6: 図 5.1 の CP-ABSC の一般的構成法に関する AP-sEUFCMA ゲームにおいて、攻撃者が図 5.5 と同様の攻撃を試みる場合の攻撃手順。

$D.Decap(K, C_D)$  を実行する. その実行結果は, DEM 方式  $\Pi_D$  の正当性より, 確率 1 で “ $m||\sigma$ ” となる. その後, 攻撃者は,  $D.Encap(K, m||\sigma)$  を実行し,  $C'_D$  を作成する. ここで, DEM 方式  $\Pi_D$  が一対一対応であると仮定すると, 確率 1 で  $C_D = C'_D$  が成立する.

4. 攻撃者は, Forgery フェーズで,  $C', A_s, A_d$  を出力する. 挑戦者は, 攻撃者が出力したサインクリプトテキストに対してアンサインクリプション処理を行い, 攻撃者の出力したサインクリプトテキストが正当なサインクリプトテキストであるか否かの検証を行う. ここで, 詳細なアンサインクリプション処理過程の記述は割愛するが, 検証の結果として挑戦者は “必ず” 攻撃者の出力したサインクリプトテキストは正当なサインクリプトテキストであるという判定を下す. しかし, 手順 3 の末尾で記述した通り,  $C'_D$  に関しては  $C'_D = C_D$  が成立するため, 実際には  $C' = C$  が成立している. 従って, 明らかに  $C' \in \mathcal{L}$  が成立するため, 攻撃者はこの攻撃によってゲームに勝利することはできない.

従って, 図 5.1 の構成法  $\Pi_{CS}$  が, AP-sEUF-CMA 安全性を達成する上で, その達成を阻む自明な攻撃法は存在しない.

また, 図 5.4 の攻撃法は, 単純な Sign-then-Encrypt 式構成法  $\Pi_{CS(StE)}$ , 及び同じく Sign-then-Encrypt 式構成法の一つである図 5.1 の構成法  $\Pi_{CS}$  に対しては, 成功しないことは自明である. 従って, AP-sEUF-CMA と同様, 図 5.1 の構成法  $\Pi_{CS}$  が, AP-IND-CCA 安全性を達成する上で, その達成を阻む自明な攻撃法は存在しない.

以上より, 図 5.1 の構成法  $\Pi_{CS}$  に関しては, AP-IND-CCA と AP-sEUF-CMA 等の最強の安全性を達成する上で, その達成を阻む自明な攻撃法は存在しないことが判明したため, 構成法  $\Pi_{CS}$  に関して安全性証明を厳密に行う価値があると判断した. そして, 次節の 5.3 節で示す通り, 安全性証明を厳密に行い, “復号者アクセス構造開示性”(CP-ABE, CP-ABKEM, CP-ABSC の性質), “署名者アクセス構造衝突困難性”(SP-ABS の性質) を新たな性質として独自に定義し安全性を証明するための仮定としてそれらを加えることで, 図 5.1 の構成法  $\Pi_{CS}$  は AP-IND-CCA, AP-sEUF-CMA, 完全匿名性という最強の安全性を達成することを示す事ができた.

### 5.3 安全性証明

図 5.1 の CP-ABSC 方式の一般的構成法  $\Pi_{CS}$  の安全性は定理 5.1 及び定理 5.2, 定理 5.3 によって保障される.

**定理 5.1.** CP-ABKEM 方式  $\Pi_{CK}$  が AP-IND-CCA 安全であり, かつ DEM 方式  $\Pi_D$  が IND-CCA 安全であり, かつ SP-ABS 方式  $\Pi_{SS}$  が署名者アクセス構造衝突困難ならば, 図 5.1 の CP-ABSC 方式  $\Pi_{CS}$  は AP-IND-CCA 安全である.

**定理 5.2.**  $SP\text{-}ABS$  方式  $\Pi_{SS}$  が  $AP\text{-}sEUF\text{-}CMA$  安全であり, かつ  $DEM$  方式  $\Pi_D$  が 1 対 1 対応であり, かつ  $CP\text{-}ABKEM$  方式が復号者アクセス構造開示的ならば, 図 5.1 の  $CP\text{-}ABSC$  方式  $\Pi_{CS}$  は  $AP\text{-}sEUF\text{-}CMA$  安全である.

**定理 5.3.**  $SP\text{-}ABS$  方式  $\Pi_{SS}$  が完全匿名ならば, 図 5.1 の  $CP\text{-}ABSC$  方式  $\Pi_{CS}$  は完全匿名である.

**定理 5.1 の証明**  $\text{Game}_0, \text{Game}_1, \text{Game}_2$  の安全性ゲームを以下のように定義する.

**Game<sub>0</sub>:** PPTA 攻撃者  $\mathcal{A}$  と挑戦者  $\mathcal{CH}$  との間で行われる,  $\Pi_{CS}$  に関する  $AP\text{-}IND\text{-}CCA$  安全性ゲーム.

**Game<sub>1</sub>:**  $\text{Game}_0$  に以下の変更を加えたゲームとする.

- Query Phase 2 のアンサインクリプションオラクルにおいて,  $\mathcal{A}$  が発行するクエリ  $(C = (C_K, C_D), A_s, S_r)$  に関して,  $C_K = C_K^*$  かつ  $C_D = C_D^*$  かつ  $A_s \neq A_s^*$  かつ  $S_r \in \mathbb{A}_d^*$  であれば,  $\mathcal{CH}$  は  $\perp$  を返答する.

**Game<sub>2</sub>:**  $\text{Game}_1$  に以下の変更を加えたゲームとする.

- Setup Phase において,  $\mathcal{CH}$  は  $K' \stackrel{U}{\leftarrow} \mathcal{K}$  を実行する.
- Challenge Phase において,  $\mathcal{CH}$  はチャレンジサインクリプトテキスト  $C^* = (C_K^*, C_D^*)$  の第二成分  $C_D^*$  を鍵  $K'$  を用いて生成する.
- Query Phase 2 のアンサインクリプションオラクルにおいて,  $\mathcal{A}$  が発行するクエリ  $(C = (C_K, C_D), A_s, S_r)$  に関して,  $C_K = C_K^*$  かつ  $C_D \neq C_D^*$  かつ  $S_r \in \mathbb{A}_d^*$  であれば,  $\mathcal{CH}$  は,  $C_D$  を鍵  $K'$  で復号する.

$\text{Game}_i (i = \{0, 1, 2\})$  の Guess Phase で攻撃者  $\mathcal{A}$  が正しい推測ビット  $b' = b$  を出力する事象を  $W_i$  と表記する.  $\text{Game}_0$  における攻撃者  $\mathcal{A}$  の優位性は定義より,

$$\begin{aligned} \text{Adv}_{\Pi_{CS}, \mathcal{A}}^{\text{AP-IND-CCA}} &= |\Pr[W_0] - \frac{1}{2}| \\ &\leq |\Pr[W_0] - \Pr[W_1]| + |\Pr[W_1] - \Pr[W_2]| + |\Pr[W_2] - \frac{1}{2}| \quad (5.1) \end{aligned}$$

不等式 (5.1) と以下で証明する補題 5.1.1, 補題 5.1.2, 補題 5.1.3 より,  $\Pi_{CK}$  が  $AP\text{-}IND\text{-}CCA$  安全であり, かつ  $\Pi_D$  が  $IND\text{-}CCA$  安全であり, かつ  $\Pi_{SS}$  が署名者アクセス構造衝突困難ならば, いかなる PPTA 攻撃者  $\mathcal{A}$  に対しても,  $\text{Adv}_{\Pi_{CS}, \mathcal{A}}^{\text{AP-IND-CCA}}$  はセキュリティパラメータ  $k$  に関して無視できるほど小さい値になる. ゆえに, 定理 5.1 が成立する.  $\square$

**補題 5.1.1.**  $SP\text{-}ABS$  方式  $\Pi_{SS}$  が署名者アクセス構造衝突困難ならば, いかなる PPTA 攻撃者  $\mathcal{A}$  に対しても,  $|\Pr[W_0] - \Pr[W_1]|$  はセキュリティパラメータ  $k$  に関して無視できるほど小さい値になる.

**補題 5.1.2.**  $CP$ - $ABKEM$  方式  $\Pi_{CK}$  が  $AP$ - $IND$ - $CCA$  安全ならば, いかなる  $PPTA$  攻撃者  $\mathcal{A}$  に対しても,  $|\Pr[W_1] - \Pr[W_2]|$  はセキュリティパラメータ  $k$  に関して無視できるほど小さい値になる.

**補題 5.1.3.**  $DEM$  方式  $\Pi_D$  が  $IND$ - $CCA$  安全ならば, いかなる  $PPTA$  攻撃者  $\mathcal{A}$  に対しても,  $|\Pr[W_2] - \frac{1}{2}|$  は無視できるほど小さい値になる.

以降では, 補題 5.1.1, 補題 5.1.2, 補題 5.1.3 の証明を行う.

**補題 5.1.1 の証明** 以下に  $\text{Game}_0$  における攻撃者  $\mathcal{A}$  と挑戦者  $\mathcal{CH}$  の具体的なやり取りを示す.

**Setup Phase:**  $\mathcal{CH}$  は  $\text{SS.Setup}(1^k, \mathcal{U}_s) \rightarrow (\text{PK}_{ss}, \text{MK}_{ss})$ ,  $\text{CK.Setup}(1^k, \mathcal{U}_r) \rightarrow (\text{PK}_{ck}, \text{MK}_{ck})$  を実行し,  $\text{PK} := (\text{PK}_{ss}, \text{PK}_{ck})$  を  $\mathcal{A}$  に送る.

**Query Phase 1:**  $\mathcal{A}$  から  $\mathcal{CH}$  への各オラクルクエリに対する  $\mathcal{CH}$  の動作は以下の通りである.

**送信者用秘密鍵生成:**  $\mathcal{A}$  が属性集合  $S_s$  をクエリする.  $\mathcal{CH}$  は  $\text{SS.KeyGen}(\text{PK}_{ss}, \text{MK}_{ss}, S_s) \rightarrow \text{SK}_s$  を実行し  $\text{SK}_s$  を  $\mathcal{A}$  へ返す.

**受信者用秘密鍵生成:**  $\mathcal{A}$  が属性集合  $S_r$  をクエリする.  $\mathcal{CH}$  は  $\text{CK.KeyGen}(\text{PK}_{ck}, \text{MK}_{ck}, S_r) \rightarrow \text{SK}_r$  を実行し  $\text{SK}_r$  を  $\mathcal{A}$  に送る.

**サインクリプション:**  $\mathcal{A}$  が平文  $m$ , 署名者アクセス構造  $\mathbb{A}_s$ , 復号者アクセス構造  $\mathbb{A}_d$ , 属性集合  $S_s$  をクエリする.  $\mathcal{CH}$  は以下の処理を実行する.  $\text{SS.KeyGen}(\text{PK}_{ss}, \text{MK}_{ss}, S_s) \rightarrow \text{SK}$ ,  $\text{CK.Encap}(\text{PK}_{ck}, \mathbb{A}_d) \rightarrow (K, C_K)$ ,  $\text{SS.Sig}(\text{PK}_{ss}, m \| C_K, \text{SK}_s'', \mathbb{A}_s) \rightarrow \sigma$ ,  $\text{D.Encap}(K, m \| \sigma) \rightarrow C_D$ . そして,  $C := (C_K, C_D)$  を  $\mathcal{A}$  へ返す.

**アンサインクリプション:**  $\mathcal{A}$  がサインクリプテキスト  $C = (C_K, C_D)$ , 署名者アクセス構造  $\mathbb{A}_s$ , 属性集合  $S_r$  をクエリする.  $\mathcal{CH}$  は  $\text{CK.KeyGen}(\text{PK}_{ck}, \text{MK}_{ck}, S_r) \rightarrow \text{SK}_r$ ,  $\text{CK.Decap}(\text{PK}_{ck}, C_K, \text{SK}_r) =: x$  を実行する.  $x \neq \perp$  なら,  $S$  は  $K := x$ ,  $\text{D.Decap}(K, C_D) =: y$  を実行する.  $y \neq \perp$  なら,  $\mathcal{CH}$  は  $m \| \sigma := y$ ,  $\text{SS.Ver}(\text{PK}_{ss}, \sigma, m \| C_K, \mathbb{A}_s) =: z$  を実行し,  $z = 1$  なら,  $m$  を  $\mathcal{A}$  へ返す.  $x = \perp$  または  $y = \perp$  または  $z = 0$  なら,  $\perp$  を返す.

**Challenge Phase:**  $\mathcal{A}$  が長さの等しい平文  $m_0, m_1$ , ターゲット署名者アクセス構造  $\mathbb{A}_s^*$ , ターゲット復号者アクセス構造  $\mathbb{A}_d^*$ , ターゲット送信者属性集合  $S_s^*$  を送る.  $\mathcal{CH}$  は以下の処理を実行する.  $\text{SS.KeyGen}(\text{PK}_{ss}, \text{MK}_{ss}, S_s^*) \rightarrow \text{SK}$ ,  $b \stackrel{\text{U}}{\leftarrow} \{0, 1\}$ ,  $\text{CK.Encap}(\text{PK}_{ck}, \mathbb{A}_d^*) \rightarrow (K^*, C_K^*)$ ,  $\text{SS.Sig}(\text{PK}_{ss}, m_b \| C_K^*, \text{SK}_s'', \mathbb{A}_s^*) \rightarrow \sigma^*$ ,  $\text{D.Decap}(K^*, m_b \| \sigma^*) \rightarrow C_D^*$ . 結果的に  $\mathcal{CH}$  は  $C^* := (C_K^*,$

$C_D^*$ ) を  $\mathcal{A}$  へ送る.

**Query Phase 2:**  $\mathcal{A}$  から  $\mathcal{CH}$  への各オラクルクエリに対する  $\mathcal{CH}$  の動作は以下の通りである.

送信者用秘密鍵生成: Query Phase 1 と同じ.

受信者用秘密鍵生成: Query Phase 1 と同じ.

サインクリプション: Query Phase 1 と同じ.

アンサインクリプション: Query Phase 1 と同じ.

**Guess Phase:**  $\mathcal{A}$  はチャレンジビット  $b$  に対する推測として  $b'$  を  $\mathcal{CH}$  へ送る.

$\text{Game}_0$  と  $\text{Game}_1$  の違いは,  $\mathcal{A}$  が Query Phase 2 においてアンサインクリプションオラクルに,  $C_K = C_K^*$  かつ  $C_D = C_D^*$  かつ  $A_s \neq A_s^*$  かつ  $S_r \in A_d^*$  を満たす ( $C = (C_K, C_D), A_s, S_r$ ) をクエリする所のみ, 現れる可能性がある. 具体的には,  $\text{Game}_0$  において  $\mathcal{A}$  が, Query Phase 2 のアンサインクリプションオラクルに,  $C_K = C_K^*$  かつ  $C_D = C_D^*$  かつ  $A_s \neq A_s^*$  かつ  $S_r \in A_d^*$  を満たすような ( $C = (C_K, C_D), A_s, S_r$ ) をクエリし, かつ  $\mathcal{CH}$  がそのクエリに対してアンサインクリプション結果として非  $\perp$  を返答する場合, この状況で必ず  $\perp$  が返答される  $\text{Game}_1$  との違いが生じる. ここで, 補題 5.1.1.1 を以下の通りに定義すると, 補題 5.1.1.1 が成立するならば, 補題 5.1.1 は明らかに成立する. つまり, いかなる PPTA 攻撃者  $\mathcal{A}$  に対しても,  $|\Pr[W_0] - \Pr[W_1]|$  はセキュリティパラメータ  $k$  に関して無視できるほど小さい値になる.  $\square$

**補題 5.1.1.1.** 事象  $P$  を次の通りに定義すると,  $\Pi_{\text{SS}}$  が署名者アクセス構造衝突困難性を備えるならば, 事象  $P$  の生起確率  $\Pr[P]$  は無視できるほど小さい値になる.

事象  $P$

$\text{Game}_0$  において, Query Phase 2 のアンサインクリプションオラクルで,  $\mathcal{A}$  が  $C_K = C_K^*$  かつ  $C_D = C_D^*$  かつ  $A_s \neq A_s^*$  かつ  $S_r \in A_d^*$  を満たし, かつアンサインクリプション結果が非  $\perp$  になるような, ( $C = (C_K, C_D), A_s, S_r$ ) を少なくとも一回クエリする. 言い換えれば,  $\mathcal{A}$  が次の式を満たすようなクエリを発行する.  $\text{CS.KeyGen}_R(\text{PK}, \text{MK}, S_r) \rightarrow \text{SK}_r, \text{CS.USC}(\text{PK}, (C_K, C_D), \text{SK}_r, A_s) \neq \perp$ .

**補題 5.1.1.1 の証明** 事象  $P$  で言及したようなアンサインクリプションクエリを  $\mathcal{A}$  が発行した場合に  $\mathcal{CH}$  が行うアンサインクリプション処理について具体的に考える.

$\mathcal{A}$  が  $C_K = C_K^*$  かつ  $C_D = C_D^*$  かつ  $A_s \neq A_s^*$  かつ  $S_r \in A_d^*$  を満たす ( $C = (C_K, C_D), A_s, S_r$ ) を Query Phase 2 のアンサインクリプションオラクルへクエリした場合, まず  $\mathcal{CH}$  は  $\text{CK.KeyGen}(\text{PK}_{ck}, \text{MK}_{ck}, S_r) \rightarrow \text{SK}_r, \text{CK.Decap}(\text{PK}_{ck}, C_K^*, \text{SK}_r) =: x$  を実行する. ここで, CP-ABKEM 方式  $\Pi_{\text{CK}}$  の正当性より, 必ず  $x = K^*$  になる. 続けて,  $\mathcal{CH}$  は  $\text{D.Decap}(K^*, C_D^*) =: y$  を実行する. ここで, DEM 方式  $\Pi_D$  の正当性より, 必ず

$y = m_b \parallel \sigma^*$  になる. 続けて,  $\mathcal{CH}$  は  $\text{SS.Ver}(\text{PK}_{ss}, \sigma^*, m_b \parallel C_K^*, \mathbb{A}_s) =: z$  を実行する. 最終的に  $\mathcal{CH}$  は  $z = 0$  であれば  $\perp$  を返答し,  $z = 1$  であれば  $m_b$  を返答する. 事象  $\mathbf{P}'$  を次のように定義すると, 以上の議論から事象  $\mathbf{P}$  と事象  $\mathbf{P}'$  に関して両者の生起確率の間に以下の関係式が成立する.

$$\Pr[\mathbf{P}] = \Pr[\mathbf{P}'] \quad (5.2)$$

事象  $\mathbf{P}'$

$\text{Game}_0$  において, Query Phase 2 のアンサインクリプションオラクルで,  $\mathcal{A}$  が  $C_K = C_K^*$  かつ  $C_D = C_D^*$  かつ  $\mathbb{A}_s \neq \mathbb{A}_s^*$  かつ  $S_r \in \mathbb{A}_d^*$  を満たし, かつ  $\text{SS.Ver}(\text{PK}_{ss}, \sigma^*, m_b \parallel C_K^*, \mathbb{A}_s) = 1$  を満たす,  $(C = (C_K, C_D), \mathbb{A}_s, S_r)$  を少なくとも一回クエリする.

式 (5.2), 補題 5.1.1.1.1 より,  $\Pi_{SS}$  が署名者アクセス構造衝突困難性を備えるならば,  $\Pr[\mathbf{P}]$  は無視できるほど小さい値になる. ゆえに, 補題 5.1.1.1 が成立する.  $\square$

**補題 5.1.1.1.** *SP-ABS* 方式  $\Pi_{SS}$  が署名者アクセス構造衝突困難性を備えるならば, いかなる *PPTA*  $\mathcal{A}$  に対しても,  $\Pr[\mathbf{P}']$  は無視できるほど小さい値になる.

以降では, 補題 5.1.1.1.1 の証明を示す.

**補題 5.1.1.1 の証明** *PPTA*  $\mathcal{A}$  は  $\text{Game}_0$  の攻撃者であるとする. また, シミュレータ  $\mathcal{S}$  は  $\mathcal{A}$  に対し,  $\text{Game}_0$  を完璧にシミュレートし, 自身は  $\Pi_{SS}$  に関する署名者アクセス構造衝突困難性ゲームの攻撃者として動作する. また,  $\mathcal{CH}$  は  $\Pi_{SS}$  に関する署名者アクセス構造衝突困難性ゲームの挑戦者を表す. なお,  $\Pi_{SS}$  は署名者アクセス構造衝突困難性を備えるとする. さて,  $\mathcal{S}$  は以下の通りに動作するものとする.

**Setup Phase:**  $\mathcal{S}$  は  $\mathcal{CH}$  より  $\text{PK}_{ss}$  を受け取る. そして,  $\text{CK.Setup}(1^k, \mathcal{U}_r) \rightarrow (\text{PK}_{ck}, \text{MK}_{ck})$  を実行し,  $\text{PK} := (\text{PK}_{ss}, \text{PK}_{ck})$  を  $\mathcal{A}$  に送る.

**Query Phase 1:**  $\mathcal{A}$  が発行する各オラクルクエリに対する  $\mathcal{S}$  の動作は以下の通りである.

**送信者用秘密鍵生成:**  $\mathcal{A}$  が属性集合  $S_s$  をクエリする.  $\mathcal{S}$  は  $\mathcal{CH}$  へ秘密鍵生成オラクルクエリとして  $S_s$  を  $\mathcal{CH}$  へ送り,  $\text{SK}_{S_s}$  を受け取り,  $\text{SK}_{S_s}$  を  $\mathcal{A}$  へ送る.

**受信者用秘密鍵生成:**  $\mathcal{A}$  が属性集合  $S_r$  をクエリする.  $\mathcal{S}$  は  $\text{CK.KeyGen}(\text{PK}_{ck}, \text{MK}_{ck}, S_r) \rightarrow \text{SK}_r$  を実行し,  $\text{SK}_r$  を  $\mathcal{A}$  に送る.

**サインクリプション:**  $\mathcal{A}$  が平文  $m$ , 署名者アクセス構造  $\mathbb{A}_s$ , 復号者アクセス構造  $\mathbb{A}_d$ , 属性集合  $S_s$  をクエリする.  $\mathcal{S}$  は  $\mathcal{CH}$  へ秘密鍵生成オラクルクエリとして  $S_s$  を送り,  $\text{SK}_{S_s}$  を受け取り, 以下の処理を実行する.  $\text{CK.Encap}(\text{PK}_{ck}, \mathbb{A}_d) \rightarrow (K, C_K)$ ,  $\text{SS.Sig}(\text{PK}_{ss}, m \parallel C_K, \text{SK}_{S_s}, \mathbb{A}_s) \rightarrow$

$\sigma$ ,  $D.\text{Encap}(K, m \parallel \sigma) \rightarrow C_D$ . そして,  $C := (C_K, C_D)$  を  $\mathcal{A}$  へ返す.

**アンサインクリプション:**  $\mathcal{A}$  がサインクリプテキスト  $C = (C_K, C_D)$ , 署名者アクセス構造  $A_s$ , 属性集合  $S_r$  をクエリする.  $S$  は  $CK.\text{KeyGen}(PK_{ck}, MK_{ck}, S_r) \rightarrow SK_r$ ,  $CK.\text{Decap}(PK_{ck}, C_K, SK_r) =: x$  を実行する.  $x \neq \perp$  ならば,  $S$  は  $K := x$ ,  $D.\text{Decap}(K, C_D) =: y$  を実行する.  $y \neq \perp$  ならば,  $S$  は  $m \parallel \sigma := y$ ,  $SS.\text{Ver}(PK_{ss}, \sigma, m \parallel C_K, A_s) =: z$  を実行する.  $S$  は  $z = 1$  ならば,  $m$  を  $\mathcal{A}$  へ送り,  $x = \perp$  または  $y = \perp$  または  $z = 0$  ならば,  $\perp$  を  $\mathcal{A}$  へ送る.

**Challenge Phase:**  $\mathcal{A}$  が長さの等しい平文  $m_0, m_1$ , ターゲット署名者アクセス構造  $A_s^*$ , ターゲット復号者アクセス構造  $A_d^*$ , ターゲット送信者属性集合  $S_s^*$  を送る.  $S$  は  $b \xleftarrow{U} \{0, 1\}$ ,  $CK.\text{Encap}(PK_{ck}, A_d^*) \rightarrow (K^*, C_K^*)$  を実行する. 続けて,  $S$  は  $CH \leftarrow (m_b \parallel C_K^*, S_s^*, A_s^*)$  を送り,  $\sigma^*$  を受け取る. 続けて,  $S$  は,  $D.\text{Encap}(K^*, m_b \parallel \sigma^*) \rightarrow C_D^*$  を実行し,  $C^* := (C_K^*, C_D^*)$  を  $\mathcal{A}$  へ返す.

**Query Phase 2:**  $\mathcal{A}$  から  $S$  への各オラクルクエリに対する  $S$  の動作は以下の通りである.

送信者用秘密鍵生成: Query Phase 1 と同じ.

受信者用秘密鍵生成: Query Phase 1 と同じ.

サインクリプション: Query Phase 1 と同じ.

アンサインクリプション:  $\mathcal{A}$  が  $(C_K, C_D)$ ,  $A_s$ ,  $S_r$  をクエリする.

(I)  $C_K = C_K^*$ , かつ  $C_D = C_D^*$ , かつ  $A_s \neq A_s^*$ , かつ  $S_r \in A_d^*$  の場合,  $S$  は  $SS.\text{Ver}(PK_{ss}, \sigma^*, m_b \parallel C_K^*, A_s) =: z$  を実行する.  $z = 1$  ならば,  $S$  は  $CH$  に対して, 署名者アクセス構造衝突困難性ゲームの Output Phase で  $A_s$  を出力して,  $\mathcal{A}$  に対する  $\text{Game}_0$  のシミュレーションはこの時点で強制的に終了する.  $z = 0$  ならば,  $S$  は  $\mathcal{A}$  に  $\perp$  を返す.

(II) その他の場合,  $S$  は Query Phase 1 での動作と同じ動作を行う.

**Guess Phase:**  $\mathcal{A}$  はチャレンジビット  $b$  に対する推測として  $b'$  を  $CH$  へ送る.

$S$  は  $\mathcal{A}$  に対し,  $\text{Game}_0$  を完璧にシミュレートできていることは自明であり, 詳細な説明は割愛する. この状況で事象  $P'$  が生じた場合に  $S$  が署名者アクセス構造衝突困難性ゲームに勝利できることは自明である. 従って, 次の等式が成立する.

$$\Pr[P'] = \text{Adv}_{\Pi_{SS}, S}^{\text{SASCR}} \quad (5.3)$$

式 (5.3) より,  $\Pr[P']$  が無視できなくなると仮定すると,  $\text{Adv}_{\Pi_{SS}, S}^{\text{SASCR}}$  が無視できなくなる. だが, これは  $\Pi_{SS}$  が署名者アクセス構造衝突困難性を備えるという事実と矛盾

しているため、背理法より仮定が誤りである。従って、補題 5.1.1.1 は成立する。□

**補題 5.1.2 の証明** 補題 5.1.2 の証明内においては  $\mathcal{CH}$  は CP-ABKEM 方式  $\Pi_{\text{CK}}$  に関する AP-IND-CCA 安全性ゲームにおける挑戦者を意味するものとする。なお、 $\Pi_{\text{CK}}$  は AP-IND-CCA 安全であるとする。PPTA 攻撃者  $\mathcal{A}$  は  $\text{Game}_1$ ,  $\text{Game}_2$  それぞれのゲームにおける攻撃者として動作する。シミュレータ  $\mathcal{S}$  は  $\Pi_{\text{CK}}$  に関する AP-IND-CCA ゲームにおける攻撃者として動作する。 $\mathcal{S}$  は  $\Pi_{\text{CK}}$  に関する AP-IND-CCA ゲームにおいて  $\mathcal{CH}$  側で決定されるチャレンジビット  $\beta$  は知らない状況で、チャレンジ鍵及びチャレンジ鍵暗号文  $(K_\beta, C_K^*)$  を利用して、 $\mathcal{A}$  に対して、 $\beta = 1$  の場合に  $\text{Game}_1$  を、 $\beta = 0$  の場合に  $\text{Game}_2$  を完全に正しくシミュレートしたい。そのために、 $\mathcal{S}$  は以下のように動作する。

**Setup Phase:**  $\mathcal{S}$  は  $\mathcal{CH}$  から  $\text{PK}_{ck}$  を受取る。 $\mathcal{S}$  は  $\text{SS.Setup}(1^k, \mathcal{U}_s) \rightarrow (\text{PK}_{ss}, \text{MK}_{ss})$ ,  $\text{PK} := (\text{PK}_{ss}, \text{PK}_{ck})$  を実行し、 $\text{PK}$  を  $\mathcal{A}$  へ送る。

**Query Phase 1:**  $\mathcal{A}$  から  $\mathcal{S}$  への各オラクルクエリに対する  $\mathcal{S}$  の動作は以下の通りである。

**送信者用秘密鍵生成:**  $\mathcal{A}$  が属性集合  $S_s$  をクエリする。 $\mathcal{S}$  は  $\text{SS.KeyGen}(\text{PK}_{ss}, \text{MK}_{ss}, S_s) \rightarrow \text{SK}_s$  を実行し  $\text{SK}_s$  を  $\mathcal{A}$  へ返す。

**受信者用秘密鍵生成:**  $\mathcal{A}$  が属性集合  $S_r$  をクエリする。 $\mathcal{S}$  は  $\mathcal{CH}$  へ秘密鍵生成オラクルクエリとして  $S_r$  を送り、鍵  $\text{SK}_r$  を受け取る。 $\mathcal{S}$  は  $\mathcal{A}$  へ  $\text{SK}_r$  を返す。

**サインクリプション:**  $\mathcal{A}$  が平文  $m$ , 署名者アクセス構造  $\mathbb{A}_s$ , 復号者アクセス構造  $\mathbb{A}_d$ , 属性集合  $S_s$  をクエリする。 $\text{SK}_s := \text{SS.KeyGen}(\text{PK}_{ss}, \text{MK}_{ss}, S_s)$  として、 $\mathcal{S}$  は以下の処理を実行する。 $\text{CK.Encap}(\text{PK}_{ck}, \mathbb{A}_d) \rightarrow (K, C_K)$ ,  $\text{SS.Sig}(\text{PK}_{ss}, m \| C_K, \text{SK}_s, \mathbb{A}_s) \rightarrow \sigma$ ,  $\text{D.Encap}(K, m \| \sigma) \rightarrow C_D$ . そして、 $C := (C_K, C_D)$  を  $\mathcal{A}$  へ返す。

**アンサインクリプション:**  $\mathcal{A}$  がサインクリプテキスト  $C = (C_K, C_D)$ , 署名者アクセス構造  $\mathbb{A}_s$ , 属性集合  $S_r$  をクエリする。 $\mathcal{S}$  は  $(C_K, S_r)$  を  $\mathcal{CH}$  へ鍵復号オラクルクエリとして送り、 $x$  を受け取る。 $x \neq \perp$  なら、 $\mathcal{S}$  は  $K := x$ ,  $\text{D.Decap}(K, C_D) =: y$  を実行する。 $y \neq \perp$  なら、 $\mathcal{S}$  は  $m \| \sigma := y$ ,  $\text{SS.Ver}(\text{PK}_{ss}, \sigma, m \| C_K, \mathbb{A}_s) \rightarrow z$  を実行し、 $z = 1$  なら、 $m$  を  $\mathcal{A}$  へ返す。 $x = \perp$  または  $y = \perp$  または  $z = 0$  なら、 $\perp$  を返す。

**Challenge Phase:**  $\mathcal{A}$  が長さの等しい平文  $m_0, m_1$ , ターゲット署名者アクセス構造  $\mathbb{A}_s^*$ , ターゲット復号者アクセス構造  $\mathbb{A}_d^*$ , ターゲット送信者属性集合  $S_s^*$  を送る。 $\text{SK}_s := \text{SS.KeyGen}(\text{PK}_{ss}, \text{MK}_{ss}, S_s^*)$  として、 $\mathcal{S}$  は最初に  $b \xleftarrow{\text{U}} \{0, 1\}$  を実行する。続けて、 $\mathbb{A}_d^*$  を  $\Pi_{\text{CK}}$  に関する AP-IND-CCA ゲームのターゲット

ト復号者アクセス構造として  $\mathcal{CH}$  へ送り,  $(K_\beta, C_K^*)$  を受取る. 続けて以下の処理を実行する.  $\text{SS.Sig}(\text{PK}_{ss}, m_b \| C_K^*, \text{SK}_s, \mathbb{A}_s^*) \rightarrow \sigma^*$ ,  $\text{D.Decap}(K^*, m_b \| \sigma^*) \rightarrow C_D^*$ . 結果的に  $\mathcal{S}$  は  $C^* := (C_K^*, C_D^*)$  を  $\mathcal{A}$  へ送る.

**Query Phase 2:**  $\mathcal{A}$  から  $\mathcal{S}$  への各オラクルクエリに対する  $\mathcal{S}$  の動作は以下の通りである.

送信者用秘密鍵生成: Query Phase 1 と同じ.

受信者用秘密鍵生成: Query Phase 1 と同じ.

サインクリプション: Query Phase 1 と同じ.

アンサインクリプション:  $\mathcal{A}$  が  $(C_K, C_D)$ ,  $\mathbb{A}_s$ ,  $S_r$  をクエリする.

- (I)  $C_K = C_K^*$ , かつ  $C_D \neq C_D^*$ , かつ  $S_r \in \mathbb{A}_d^*$  の場合,  $\mathcal{S}$  は,  $\text{D.Decap}(K_\beta, C_D) := y$  を実行し,  $y \neq \perp$  ならば,  $m \| \sigma := y$ ,  $\text{SS.Ver}(\text{PK}_{ss}, \sigma, m \| C_K, \mathbb{A}_s) \rightarrow z$  を実行し,  $z = 1$  ならば  $m$  を返す.  $y = \perp$  または  $z = 0$  ならば,  $\perp$  を返す.
- (II)  $C_K = C_K^*$ , かつ  $C_D = C_D^*$ , かつ  $\mathbb{A}_s \neq \mathbb{A}_s^*$ , かつ  $S_r \in \mathbb{A}_d^*$  の場合,  $\mathcal{S}$  は  $\perp$  を返す.
- (III) その他の場合,  $\mathcal{S}$  は  $(C_K, S_r)$  を  $\mathcal{CH}$  へ鍵復号オラクルクエリとして送り,  $x$  を受け取る.  $x \neq \perp$  であれば,  $K := x$ ,  $\text{D.Decap}(K, C_D) := y$  を実行し,  $y \neq \perp$  ならば  $m \| \sigma := y$ ,  $\text{SS.Ver}(\text{PK}_{ss}, \sigma, m \| C_K, \mathbb{A}_s) := z$  を実行し,  $z = 1$  ならば  $m$  を返す.  $x = \perp$  または  $y = \perp$  または  $z = 0$  ならば,  $\perp$  を返す.

**Guess Phase:**  $\mathcal{A}$  はチャレンジビット  $b$  に対する推測ビットとして  $b'$  を  $\mathcal{S}$  へ送る.  $\mathcal{S}$  は  $b' = b$  ならば  $\beta' := 1$ ,  $b' \neq b$  ならば  $\beta' := 0$  を実行し,  $\beta'$  を  $\Pi_{\text{CK}}$  に関する AP-IND-CCA ゲームのチャレンジビット  $\beta$  に対する推測ビットとして  $\mathcal{CH}$  へ送る.

$\Pi_{\text{CK}}$  に関する AP-IND-CCA ゲームにおける  $\mathcal{S}$  の優位性は定義より,

$$\text{Adv}_{\Pi_{\text{CK}}, \mathcal{S}}^{\text{AP-IND-CCA}} = \left| \Pr[\beta' = \beta] - \frac{1}{2} \right| \quad (5.4)$$

変数  $\beta, \beta'$  に関して確率の定理より,

$$\Pr[\beta' = 0 | \beta = 0] + \Pr[\beta' = 1 | \beta = 0] = 1 \quad (5.5)$$

$$\begin{aligned} \Pr[\beta' = \beta] &= \Pr[\beta' = 0 \wedge \beta = 0] + \Pr[\beta' = 1 \wedge \beta = 1] \\ &= \Pr[\beta' = 0 | \beta = 0] \Pr[\beta = 0] + \Pr[\beta' = 1 | \beta = 1] \Pr[\beta = 1] \\ &= \frac{1}{2} (\Pr[\beta' = 0 | \beta = 0] + \Pr[\beta' = 1 | \beta = 1]) \end{aligned} \quad (5.6)$$

等式 (5.5), (5.6) より,

$$\Pr[\beta' = \beta] = \frac{1}{2}(1 - \Pr[\beta' = 1|\beta = 0] + \Pr[\beta' = 1|\beta = 1]) \quad (5.7)$$

等式 (5.7) より,

$$\begin{aligned} |\Pr[\beta' = \beta] - \frac{1}{2}| &= \frac{1}{2}|\Pr[\beta' = 1|\beta = 1] - \Pr[\beta' = 1|\beta = 0]| \\ &= \frac{1}{2}|\Pr[b' = b|\beta = 1] - \Pr[b' = b|\beta = 0]| \end{aligned} \quad (5.8)$$

等式 (5.4), (5.8) より,

$$\text{Adv}_{\Pi_{\text{CK}}, \mathcal{S}}^{\text{AP-IND-CCA}} = \frac{1}{2}|\Pr[b' = b|\beta = 1] - \Pr[b' = b|\beta = 0]| \quad (5.9)$$

$\beta = 1$  (resp.  $\beta = 0$ ) の場合に, シミュレータ  $\mathcal{S}$  が攻撃者  $\mathcal{A}$  に対して,  $\text{Game}_1$  (resp.  $\text{Game}_2$ ) の挑戦者としての応答を完璧にシミュレートできている事, そして  $\mathcal{S}$  が  $\mathcal{CH}$  に対して  $\Pi_{\text{CK}}$  に関する AP-IND-CCA ゲームのルール上禁止されているオラクルクエリを一度も行っていない事を, 説明する.

まず,  $\beta = 1$  の場合を考える.

Challenge Phase, Query Phase 2 のアンサインクリプションオラクルクエリ発行時以外に関しては,  $\mathcal{S}$  が  $\mathcal{A}$  に対して  $\text{Game}_1$  を完璧にシミュレートできている事はほぼ自明であるので, 詳細な説明は割愛する. Challenge Phase において,  $\beta = 1$  の場合,  $(K_\beta, C_K^*)$  は  $\mathcal{CH}$  側で  $\text{CK.Encap}(\text{PK}_{\text{ck}}, A_d^*)$  の出力結果として生成されており, 鍵及び鍵暗号文生成以降の動作は,  $\mathcal{S}$  が行う動作と  $\text{Game}_1$  の挑戦者が行う動作は全く同じであるので,  $\mathcal{A}$  から見て  $\mathcal{S}$  が生成するチャレンジサインクリプトテキストと  $\text{Game}_1$  の挑戦者が生成するチャレンジサインクリプトテキストは全く見分けがつかない. 従って,  $\mathcal{S}$  は Challenge Phase において  $\text{Game}_1$  を正しくシミュレート出来ている. Query Phase 2 のアンサインクリプションオラクルクエリが発行される時, (I) のタイプのクエリの場合は,  $C_K = C_K^*$ , かつ  $S_r \in A_d^*$  だから,  $\text{CK.KeyGen}(\text{PK}_{\text{ck}}, S_r) := \text{SK}_r$  とすると, CP-ABKEM 方式  $\Pi_{\text{CK}}$  の正当性より確率 1 で,  $\text{CK.Decap}(\text{PK}_{\text{ck}}, C_K, \text{SK}_r) = K_\beta = K_1$ , となる. 従って,  $\mathcal{S}$  が  $\mathcal{A}$  からのクエリに関して,  $K_\beta$  を利用してアンサインクリプションした結果は,  $\text{Game}_1$  の挑戦者が同一のクエリに関してアンサインクリプションした結果と, 必ず同一になる. 従って,  $\mathcal{S}$  は (I) の場合  $\text{Game}_1$  を正しくシミュレートできている. 続いて (II) の場合は,  $\mathcal{S}$  は  $\text{Game}_1$  の挑戦者が行うべき応答を完璧にシミュレートできていることは明らかである. 最後に (III) の場合は,  $C_K \neq C_K^*$  または  $S_r \notin A_d^*$  のいずれかが必ず成立するので,  $\mathcal{S}$  が  $\mathcal{CH}$  に鍵復号オラクルクエリとして  $(C_K, S_r)$  を発行する事は禁止ではない. 従って,  $\mathcal{S}$  が正しい鍵復号結果を  $\mathcal{CH}$  から受取り, それを用いてアンサインクリプション処理を継続し, 最終的に導き出す結果は正しいアンサインクリプション結果である. 従って, (III) の場合  $\mathcal{S}$  は  $\text{Game}_1$  の返答を正しくシミュレートできている. 以上より,  $\mathcal{S}$  は  $\beta = 1$  の場合, ゲーム開始から終了まで一貫して  $\mathcal{A}$  に対して  $\text{Game}_1$  の挑戦者としての応答を完璧にシミュレート出来ている.

次に  $\beta = 1$  の場合に  $\mathcal{S}$  が  $\mathcal{CH}$  に対して禁止されたオラクルクエリを一度も行っていない事を説明する.  $\mathcal{S}$  が禁止のクエリを行う可能性があるオラクルは, 3つあり,  $\mathcal{S}$  がそのオラクルクエリを行う契機となる  $\mathcal{A}$  からのクエリが行われるオラクルと共に以下に示す. (1)Query Phase 1 の秘密鍵生成オラクル (Query Phase 1 の受信者用秘密鍵生成オラクル), (2)Query Phase 2 の秘密鍵生成オラクル (Query Phase 2 の受信者用秘密鍵生成オラクル), (3)Query Phase 2 の鍵復号オラクル (Query Phase 2 のアンサインクリプションオラクル). (1)に関して,  $\mathcal{S}$  にとって  $S_r \in \mathbb{A}_d^*$  を満たす  $S_r$  を Query Phase 1 の秘密鍵生成オラクルでクエリする事は禁止されている. 一方で,  $\mathcal{A}$  にとっても  $S_r \in \mathbb{A}_d^*$  を満たす  $S_r$  を Query Phase 1 の受信者用秘密鍵生成オラクルでクエリする事は禁止されている. 本証明においては  $\mathcal{A}$  が禁止されたオラクルクエリを行う事はないと仮定できるため,  $\mathcal{A}$  が  $\mathcal{S}$  に対し  $S_r \in \mathbb{A}_d^*$  を満たす  $S_r$  をクエリする事は一度もない. 従って, (1)において  $\mathcal{S}$  が  $\mathcal{CH}$  に対し禁止されたオラクルクエリを行う事は一度もない. (2)に関しては, (1) と全く同一の理由で,  $\mathcal{S}$  が  $\mathcal{CH}$  に対し禁止されたオラクルクエリを行う事は一度もない. (3)に関して,  $\mathcal{S}$  にとって,  $C_K = C_K^*$ , かつ  $S_r \in \mathbb{A}_d^*$  を満たす  $(C_K, S_r)$  を,  $\Pi_{\text{CK}}$  の AP-IND-CCA ゲームの Query Phase 2 で, 鍵復号オラクルクエリとして  $\mathcal{CH}$  へ送る事は禁止されている.  $\mathcal{A}$  が Query Phase 2 のアンサインクリプションオラクルで  $((C_K, C_C), \mathbb{A}_s, S_r)$  をクエリし, このクエリが  $C_K = C_K^*$  かつ  $S_r \in \mathbb{A}_d^*$  を満たす場合, 他の変数に関して以下の4通りの可能性が存在する. (i) $C_D = C_D^*$  かつ  $\mathbb{A}_s = \mathbb{A}_s^*$ , (ii) $C_D = C_D^*$  かつ  $\mathbb{A}_s \neq \mathbb{A}_s^*$ , (iii) $C_D \neq C_D^*$  かつ  $\mathbb{A}_s = \mathbb{A}_s^*$ , (iv) $C_D \neq C_D^*$  かつ  $\mathbb{A}_s \neq \mathbb{A}_s^*$ . これらの内, (i)に関してはゲームのルール上禁止されたクエリであるため, このようなクエリを  $\mathcal{A}$  が発行することはない. (ii)のタイプのクエリが行われた場合,  $\mathcal{S}$  は (II) の対応をとる. (iii), (iv)のタイプのクエリが行われた場合,  $\mathcal{S}$  は (I) の対応をとる. そして,  $\mathcal{S}$  が (I), (II) の対応をとる場合, いずれの対応においても,  $\mathcal{S}$  は鍵復号オラクルを利用しない.  $\mathcal{S}$  が鍵復号オラクルを利用するのは (III) の対応をとる場合のみであり, 以上の議論から (III) の対応をとる場合には  $C_K \neq C_K^*$  または  $S_r \notin \mathbb{A}_d^*$ . 従って, (3)において  $\mathcal{S}$  が  $\mathcal{CH}$  に対し禁止のオラクルクエリを行う事は一度もない. ゆえに,  $\beta = 1$  の場合に,  $\mathcal{S}$  が  $\mathcal{CH}$  に対して禁止のオラクルクエリを行う事は一度もない.

次に  $\beta = 0$  の場合を考える.

Challenge Phase, または Query Phase 2 のアンサインクリプションオラクルにおいて (I) のタイプのクエリが発行される時, 以外に関しては,  $\mathcal{S}$  が  $\mathcal{A}$  に対して  $\text{Game}_2$  を完璧にシミュレートできていることは,  $\beta = 1$  の場合と同様に, ほぼ自明なので, 詳細な説明は割愛する. Challenge Phase においては,  $\beta = 0$  の場合,  $K_\beta = K_0$  は鍵空間  $\mathcal{K}$  よりランダムに生成されたものであり,  $\mathcal{S}$  はこの鍵  $K_0$  を利用してチャレンジサインクリプトテキストを生成している. そのため,  $\mathcal{A}$  が  $\mathcal{S}$  が生成するチャレンジサインクリプトテキストと  $\text{Game}_2$  において挑戦者が生成するチャレンジサインクリプトテキストとを見分ける事は不可能. 従って, Challenge Phase において,  $\mathcal{S}$  は  $\text{Game}_2$  を正しくシミュレート出来ている. 次に, Query Phase 2 のアンサインクリプションオラクルで  $\mathcal{A}$  が (I) のタイプのクエリを行った場合に関して,  $\mathcal{S}$  は  $\mathcal{K}$  よりランダムに選ばれた  $K_0$  を利用して  $C_D$  の復号を行っているので,  $\mathcal{S}$  は明らかに  $\text{Game}_2$  で挑戦者が行う返答を正しくシミュレートできている. ゆえに,  $\beta = 0$  の場合,  $\mathcal{S}$  は  $\mathcal{A}$  に対し,  $\text{Game}_2$  をゲーム開始か

ら終了まで完璧にシミュレートできている。

$\beta = 0$  の場合に、 $\mathcal{S}$  が  $\mathcal{CH}$  に対して禁止されたオラクルクエリを一度も行わないことに関しては、 $\beta = 1$  の場合と全く同様の理由で説明可能であるので、詳細な説明は割愛する。

以上より、 $\beta = 1$  (resp.  $\beta = 0$ ) の場合、 $\mathcal{S}$  は  $\mathcal{A}$  に対してチャレンジビットが  $b$  であるような  $\text{Game}_1$  (resp.  $\text{Game}_2$ ) を完璧にシミュレートできていること、更に  $\mathcal{S}$  は  $\mathcal{CH}$  に対して禁止されたオラクルクエリを一度も行っていないことが確かめられた。このような状況下では、 $b' = b$  という事象の生起は事象  $W_1$  (resp.  $W_2$ ) の生起と同等であるため、以下の等式が成り立つ。

$$\Pr[b' = b | \beta = 1] = \Pr[W_1] \quad (5.10)$$

$$\Pr[b' = b | \beta = 0] = \Pr[W_2] \quad (5.11)$$

等式 (5.9), (5.10), (5.11) より、以下の等式が成り立つ。

$$\text{Adv}_{\Pi_{\text{CK}}, \mathcal{S}}^{\text{AP-IND-CCA}} = \frac{1}{2} |\Pr[W_1] - \Pr[W_2]| \quad (5.12)$$

$|\Pr[W_1] - \Pr[W_2]|$  が無視できなくなるような  $\mathcal{A}$  が存在すると仮定すると、式 (5.12) より、 $\text{Adv}_{\Pi_{\text{CK}}, \mathcal{S}}^{\text{AP-IND-CCA}}$  は無視できなくなり、これは  $\Pi_{\text{CK}}$  が AP-IND-CCA 安全であるという事実に矛盾するので、その仮定は誤りである。ゆえに、 $\Pi_{\text{CK}}$  が AP-IND-CCA 安全ならば、いかなる PPTA 攻撃者  $\mathcal{A}$  に対しても、 $|\Pr[W_1] - \Pr[W_2]|$  はセキュリティパラメータ  $k$  に関して無視できるほど小さい値になる。□

**補題 5.1.3 の証明** 補題 5.1.3 の証明内においては、 $\mathcal{CH}$  は、DEM 方式  $\Pi_{\text{D}}$  に関する IND-CCA 安全性ゲームにおける挑戦者を意味する。但し、 $\Pi_{\text{D}}$  は IND-CCA 安全であるとする。 $\mathcal{S}$  は  $\mathcal{A}$  に対して、 $\Pi_{\text{CK}}$  に関する  $\text{Game}_2$  の安全性ゲームを完璧にシミュレートし  $\mathcal{A}$  の最終的な出力を利用して  $\Pi_{\text{D}}$  に関する IND-CCA 安全性を破ろうとする PPTA であり、 $\mathcal{S}$  の動作を以下のように定める。

**Setup Phase:**  $\mathcal{S}$  は  $\text{SS.Setup}(1^k, \mathcal{U}_s) \rightarrow (\text{PK}_{ss}, \text{MK}_{ss})$ ,  $\text{CK.Setup}(1^k, \mathcal{U}_r) \rightarrow (\text{PK}_{ck}, \text{MK}_{ck})$  を計算し、 $\text{PK} := (\text{PK}_{ss}, \text{PK}_{ck})$  を  $\mathcal{A}$  に送る。

**Query Phase 1:**  $\mathcal{A}$  から  $\mathcal{S}$  への各オラクルクエリに対する  $\mathcal{S}$  の動作は以下の通りである。

**送信者用秘密鍵生成:**  $\mathcal{A}$  が属性集合  $S_s$  をクエリする。 $\mathcal{S}$  は  $\text{SS.KeyGen}(\text{PK}_{ss}, \text{MK}_{ss}, S_s) \rightarrow \text{SK}_s$  を実行し  $\text{SK}_s$  を  $\mathcal{A}$  へ返す。

**受信者用秘密鍵生成:**  $\mathcal{A}$  が属性集合  $S_r$  をクエリする。 $\mathcal{S}$  は  $\text{CK.KeyGen}(\text{PK}_{ck}, \text{MK}_{ck}, S_r) \rightarrow \text{SK}_r$  を実行し  $\text{SK}_r$  を  $\mathcal{A}$  に送る。

**サインクリプション:**  $\mathcal{A}$  が平文  $m$ , 署名者アクセス構造  $A_s$ , 復号者アクセス構造  $A_d$ , 属性集合  $S_s$  をクエリする。 $\text{SK}_s := \text{SS.KeyGen}(\text{PK}_{ss},$

$\text{MK}_{ss}, S_s$ ) として,  $\mathcal{S}$  は以下の処理を実行する.  $\text{CK.Encap}(\text{PK}_{ck}, \mathbb{A}_d) \rightarrow (K, C_K)$ ,  $\text{SS.Sig}(\text{PK}_{ss}, m \| C_K, \text{SK}_s, \mathbb{A}_s) \rightarrow \sigma$ ,  $\text{D.Encap}(K, m \| \sigma) \rightarrow C_D$ . そして,  $C := (C_K, C_D)$  を  $\mathcal{A}$  へ返す.

**アンサインクリプション:**  $\mathcal{A}$  がサインクリプテキスト  $C = (C_K, C_D)$ , 署名者アクセス構造  $\mathbb{A}_s$ , 属性集合  $S_r$  をクエリする.  $\mathcal{S}$  は  $\text{CK.KeyGen}(\text{PK}_{ck}, \text{MK}_{ck}, S_r) \rightarrow \text{SK}_r$ ,  $\text{CK.Decap}(\text{PK}_{ck}, C_K, \text{SK}_r) := x$  を実行する.  $x \neq \perp$  なら,  $\mathcal{S}$  は  $K := x$ ,  $\text{D.Decap}(K, C_D) := y$  を実行する.  $y \neq \perp$  なら,  $\mathcal{S}$  は  $m \| \sigma := y$ ,  $\text{SS.Ver}(\text{PK}_{ss}, \sigma, m \| C_K, \mathbb{A}_s) := z$  を実行し,  $z = 1$  なら,  $m$  を  $\mathcal{A}$  へ返す.  $x = \perp$  または  $y = \perp$  または  $z = 0$  なら,  $\perp$  を返す.

**Challenge Phase:**  $\mathcal{A}$  が長さの等しい平文  $m_0, m_1$ , ターゲット署名者アクセス構造  $\mathbb{A}_s^*$ , ターゲット復号者アクセス構造  $\mathbb{A}_d^*$ , ターゲット送信者属性集合  $S_s^*$  を送る.  $\mathcal{S}$  は  $\text{SK}_s := \text{SS.KeyGen}(\text{PK}_{ss}, \text{MK}_{ss}, S_s^*)$  として, 以下の処理を実行する.  $\text{CK.Encap}(\text{PK}_{ck}, \mathbb{A}_d^*) \rightarrow (K^*, C_K^*)$ ,  $\text{SS.Sig}(\text{PK}_{ss}, m_0 \| C_K^*, \text{SK}_s, \mathbb{A}_s^*) \rightarrow \sigma_0$ ,  $\text{SS.Sig}(\text{PK}_{ss}, m_1 \| C_K^*, \text{SK}_s, \mathbb{A}_s^*) \rightarrow \sigma_1$ ,  $M_0 := m_0 \| \sigma_0$ ,  $M_1 := m_1 \| \sigma_1$ .  $\mathcal{S}$  は平文  $M_0, M_1$  を  $\mathcal{CH}$  へ送り,  $C_D^*$  を受け取り,  $C^* := (C_K^*, C_D^*)$  を  $\mathcal{A}$  へ送る.

**Query Phase 2:**  $\mathcal{A}$  から  $\mathcal{S}$  への各オラクルクエリに対する  $\mathcal{S}$  の動作は以下の通りである.

**送信者用秘密鍵生成:** Query Phase 1 と同じ.

**受信者用秘密鍵生成:** Query Phase 1 と同じ.

**サインクリプション:** Query Phase 1 と同じ.

**アンサインクリプション:**  $\mathcal{A}$  が  $C = (C_K, C_D), \mathbb{A}_s, S_r$  をクエリする.

- (I)  $C_K = C_K^*$ , かつ  $C_D \neq C_D^*$ , かつ  $S_r \in \mathbb{A}_d^*$  の場合,  $\mathcal{S}$  は  $C_D$  をデータ復号オラクルクエリとして  $\mathcal{CH}$  へ送り,  $y$  を受け取る.  $\mathcal{S}$  は,  $y \neq \perp$  ならば  $m \| \sigma := y$ ,  $\text{SS.Ver}(\text{PK}_{ss}, \sigma, m \| C_K^*, \mathbb{A}_s) \rightarrow z$  を実行し,  $z = 1$  なら  $m$  を  $\mathcal{A}$  へ送る.  $y = \perp$  または  $z = 0$  なら,  $\perp$  を送る.
- (II)  $C_K = C_K^*$ , かつ  $C_D = C_D^*$ , かつ  $\mathbb{A}_s \neq \mathbb{A}_s^*$ , かつ  $S_r \in \mathbb{A}_d^*$  の場合,  $\mathcal{S}$  は  $\perp$  を返す.
- (III) その他の場合,  $\mathcal{S}$  は Query Phase 1 でアンサインクリプションオラクルクエリが発行された時の動作と同様の動作を行う.

**Guess Phase:**  $\mathcal{A}$  はチャレンジビット  $b$  に関する推測として  $b'$  を  $\mathcal{S}$  へ送る.  $\mathcal{S}$  はチャレンジビット  $\beta$  に関する推測として  $\beta' := b'$  を  $\mathcal{CH}$  へ送る.

シミュレータ  $\mathcal{S}$  が攻撃者  $\mathcal{A}$  に対して,  $\text{Game}_2$  の挑戦者としての応答を完璧にシミュレート出来ている事, そして  $\mathcal{S}$  が  $\mathcal{CH}$  に対して  $\Pi_D$  に関する IND-CCA ゲームでルー

ル上禁止されているオラクルクエリを一度も行っていない事を説明する。

まず, Challenge Phase と Query Phase 2 のアンサインクリプションオラクルで  $\mathcal{A}$  が (I) のタイプのクエリを行った場合, 以外は,  $\mathcal{S}$  が  $\text{Game}_2$  の挑戦者が行う返答を完璧にシミュレートできている事はほぼ自明であるので, 詳細な説明は割愛する. Challenge Phase においては,  $C_D^*$  は  $\mathcal{CH}$  が ( $\Pi_D$  に関する IND-CCA ゲームのチャレンジビットを  $\beta$  として)  $M_\beta = m_\beta \parallel \sigma_\beta$  を (Init フェーズで決定した鍵を  $K'$  として) 鍵  $K'$  を使って D.Encap アルゴリズムによって暗号化したものであり, 結果的に  $\mathcal{S}$  はチャレンジビットが  $b = \beta$  であるようなチャレンジサインクリプトテキストを生成して  $\mathcal{A}$  へ送っているとみなせる. 従って,  $\mathcal{S}$  は Challenge Phase においてチャレンジビットが  $b = \beta$  であるような  $\text{Game}_3$  における挑戦者の応答を完璧にシミュレート出来ている. Query Phase 2 のアンサインクリプションオラクルで  $\mathcal{A}$  が (I) のタイプのクエリを行った場合,  $\mathcal{S}$  は  $C_D$  をデータ復号オラクルクエリとして  $\mathcal{CH}$  へ送り,  $\mathcal{CH}$  が  $C_D$  を鍵  $K'$  を使って D.Decap アルゴリズムによって復号した結果を受け取る.  $\mathcal{S}$  はその復号結果を利用してアンサインクリプション処理を継続し, 最終的に得たアンサインクリプション結果を返答する. 従って,  $\mathcal{S}$  は  $\text{Game}_2$  の Query Phase 2 のアンサインクリプションオラクルで  $\mathcal{A}$  が (I) のタイプのクエリを行った場合に  $\text{Game}_2$  の挑戦者が行う返答を完璧にシミュレートできている. 以上より,  $\mathcal{S}$  はゲーム開始から終了まで  $\mathcal{A}$  に対して  $\text{Game}_2$  の挑戦者としての返答を完璧にシミュレートできている.

次に,  $\mathcal{S}$  が  $\mathcal{CH}$  に対して禁止のオラクルクエリを一度も行わないことを証明する.  $\mathcal{S}$  が  $\mathcal{CH}$  に対して禁止のオラクルクエリを行う可能性のあるタイミングは, 一か所だけ存在し, それは,  $\Pi_D$  に関する IND-CCA ゲームの Query Phase 2 のデータ復号オラクルである. そして,  $\mathcal{S}$  がこのオラクルに対してクエリを発行するきっかけは,  $\mathcal{A}$  が Query Phase 2 のアンサインクリプションオラクルで発行する (I) のタイプのクエリである. この場合, 条件より  $C_D \neq C_D^*$  であるので,  $\mathcal{S}$  が  $\mathcal{CH}$  に対して  $C_D$  をデータ復号オラクルクエリとして発行することは禁止ではない. 従って,  $\mathcal{S}$  は  $\mathcal{CH}$  に対して  $\Pi_D$  に関する IND-CCA ゲームで禁止されているオラクルクエリを一度も行わない.

ゆえに,  $\mathcal{A}$  の  $\text{Game}_2$  における優位性を  $\text{Adv}_{\Pi_{CS}, \mathcal{A}}^{\text{Game}_3}$  とすれば, 定義より,

$$\text{Adv}_{\Pi_{CS}, \mathcal{A}}^{\text{Game}_2} = |\Pr[b' = b] - \frac{1}{2}| = |\Pr[W_2] - \frac{1}{2}| \quad (5.13)$$

さらに,  $b = \beta$ ,  $b' = \beta'$  なので,  $b = b'$  ならば  $\beta = \beta'$  となるし,  $\beta = \beta'$  ならば  $b = b'$  なる. よって,  $b' = b$  という事象が生起する確率と  $\beta' = \beta$  という事象が生起する確率は等しく, 以下の等式が成り立つ.

$$\Pr[b' = b] = \Pr[\beta' = \beta] \quad (5.14)$$

DEM 方式  $\Pi_D$  に関する IND-CCA ゲームにおける  $\mathcal{S}$  の優位性の式と, 等式 (5.13), 等式 (5.14) より, 以下の等式が成り立つ.

$$\text{Adv}_{\Pi_D, \mathcal{S}}^{\text{IND-CCA}} = |\Pr[\beta' = \beta] - \frac{1}{2}| = |\Pr[b' = b] - \frac{1}{2}| = |\Pr[W_2] - \frac{1}{2}| \quad (5.15)$$

$|\Pr[W_2] - \frac{1}{2}|$  が無視できなくなるような  $\mathcal{A}$  が存在すると仮定すると, 式 (5.15) より,  $\text{Adv}_{\Pi_D, \mathcal{S}}^{\text{IND-CCA}}$  は無視できなくなり, これは  $\Pi_D$  が IND-CCA 安全であるという事実

盾するので、その仮定は誤りである。ゆえに、 $\Pi_D$  が IND-CCA 安全ならば、いかなる PPTA 攻撃者  $\mathcal{A}$  に対しても、 $|\Pr[W_2] - \frac{1}{2}|$  はセキュリティパラメータ  $k$  に関して無視できるほど小さい値になる。□

**定理 5.2 の証明** 本稿の CP-ABSC 方式の AP-sEUF-CMA 安全性の安全性定義は、当該方式が復号者アクセス構造開示性を満たすことを仮定している。従って、CP-ABSC 方式の一般的構成法  $\Pi_{CS}$  が当該安全性を達成することを示す前に、当該構成法が復号者アクセス構造開示性を満たすことを示す必要がある。CP-ABSC 方式の一般的構成法  $\Pi_{CS}$  が復号者アクセス構造開示性を満たすことに関しては、補題 5.2.1 で示す。補題 5.2.1 を踏まえて、CP-ABSC 方式の一般的構成法  $\Pi_{CS}$  が AP-sEUF-CMA 安全であることに関して、補題 5.2.2 で示す。両補題より、定理 5.2 は成立する。□

**補題 5.2.1.** CP-ABKEM 方式  $\Pi_{CK}$  が復号者アクセス構造開示性を満たすならば、図 5.1 の CP-ABSC 方式  $\Pi_{CS}$  は復号者アクセス構造開示性を満たす。

**補題 5.2.2.** SP-ABS 方式  $\Pi_{SS}$  が AP-sEUF-CMA 安全であり、DEM 方式  $\Pi_D$  が一対一対応性を満たし、CP-ABKEM 方式  $\Pi_{CK}$  が復号者アクセス構造開示性を満たすのならば、いかなる PPTA  $\mathcal{A}$  に対しても、 $\text{Adv}_{\Pi_{CS}, \mathcal{A}}^{\text{AP-sEUF-CMA}}$  はセキュリティパラメータ  $k$  に関して、無視できるほど小さい値になる。

以降には、補題 5.2.1, 補題 5.2.2 の証明を示す。

**補題 5.2.1 の証明** CP-ABKEM 方式  $\Pi_{CK}$  に関して、復号者アクセス構造開示性の定義を満たすアルゴリズムを、 $\text{Disclose}_{CK}$  と表記する。

全ての  $k$ , 全ての  $\mathcal{U}_s$ , 全ての  $\mathcal{U}_r$ , 全ての  $(\text{PK}_{ss}, \text{MK}_{ss}) \leftarrow \text{SS.Setup}(1^k, \mathcal{U}_s)$ , 全ての  $(\text{PK}_{ck}, \text{MK}_{ck}) \leftarrow \text{CK.Setup}(1^k, \mathcal{U}_r)$ , 全ての  $m$ , 全ての  $S_s \in (2^{\mathcal{U}} - \{\emptyset\})$ , 全ての  $\text{SK}_s \leftarrow \text{SS.KeyGen}(\text{PK}, \text{MK}, S_s)$ , 全ての  $\mathbb{A}_s (s.t. S_s \in \mathbb{A}_s)$ , 全ての  $\mathbb{A}_d$ , 全ての  $(K, C_K) \leftarrow \text{CK.Encap}(\text{PK}_{ck}, \mathbb{A}_d)$ , 全ての  $\sigma \leftarrow \text{SS.Sig}(\text{PK}_{ss}, m \| C_K, \text{SK}_s, \mathbb{A}_s)$ , 全ての  $C_D \leftarrow \text{D.Encap}(K, m \| \sigma)$  に対して、ここで  $\text{PK} := (\text{PK}_{ss}, \text{PK}_{ck})$ , また  $C := (C_K, C_D)$  として、アルゴリズム  $\text{Disclose}_{CK}$  をサブルーチンとして利用し、 $C$  を入力変数とするアルゴリズム  $\text{Disclose}_{CS}$  を図 5.7 のように定義する。

$\text{Disclose}_{CS}(\text{PK}, C)$  :

Parse  $\text{PK}$  as  $(\text{PK}_{ss}, \text{PK}_{ck})$ .

Parse  $C$  as  $(C_K, C_D)$ .

Return  $\text{Disclose}_{CK}(\text{PK}_{ck}, C_K)$ .

図 5.7: 図 5.1 の CP-ABSC 方式  $\Pi_{CS}$  の復号者アクセス構造開示性アルゴリズム  $\text{Disclose}_{CS}$

図 5.7 の定義より、次の等式が成立する。

$$\text{Disclose}_{CS}(\text{PK}, C) = \text{Disclose}_{CK}(\text{PK}_{ck}, C_K) \quad (5.16)$$

$\text{Disclose}_{\text{CK}}$  は  $\Pi_{\text{CK}}$  の復号者アクセス構造開示性アルゴリズムだから、CP-ABKEM方式の復号者アクセス構造開示性の定義より、次の等式が成立する.

$$\Pr[\text{Disclose}_{\text{CK}}(\text{PK}_{ck}, C_K) = \mathbb{A}_d] = 1 \quad (5.17)$$

式 (5.16), (5.17) より、次の等式が成立する.

$$\Pr[\text{Disclose}_{\text{CS}}(\text{PK}, C) = \mathbb{A}_d] = 1 \quad (5.18)$$

式 (5.18) と CP-ABSC 方式の復号者アクセス構造開示性の定義より、アルゴリズム  $\text{Disclose}_{\text{CS}}$  は復号者アクセス構造開示性アルゴリズムとしての条件を満たしている. よって、図 5.1 の CP-ABSC 方式  $\Pi_{\text{CS}}$  は復号者アクセス構造開示性を満たす. ゆえに、補題 5.2.1 は成立する.  $\square$

**補題 5.2.2 の証明** 補題 5.2.2 の証明内においては、PPTA  $\mathcal{A}$  は図 5.1 の CP-ABSC 方式  $\Pi_{\text{CS}}$  に関する AP-sEUF-CMA 安全性ゲームにおける攻撃者を意味する. 但し、 $\mathcal{A}$  は当該安全性を無視できない優位性で破ることができると仮定する. また、 $\mathcal{CH}$  は SP-ABS 方式  $\Pi_{\text{SS}}$  に関する AP-sEUF-CMA 安全性ゲームにおける挑戦者を意味する. 但し、 $\Pi_{\text{SS}}$  は AP-sEUF-CMA 安全な SP-ABS 方式であるとする.  $\mathcal{S}$  は  $\mathcal{A}$  に対して  $\Pi_{\text{CS}}$  に関する AP-sEUF-CMA 安全性ゲームを完璧にシミュレートし、 $\mathcal{A}$  の最終的な出力を利用して、 $\Pi_{\text{SS}}$  に関する AP-sEUF-CMA 安全性ゲームに勝利しようとするシミュレータを意味し、 $\mathcal{S}$  の動作を以下のように定める.

**Setup Phase:**  $\mathcal{S}$  は  $\mathcal{CH}$  より  $\text{PK}_{ss}$  を受取る.  $\mathcal{S}$  は  $\text{CK.Setup}(1^k, \mathcal{U}_r) \rightarrow (\text{PK}_{ck}, \text{MK}_{ck})$ ,  $\text{PK} := (\text{PK}_{ss}, \text{PK}_{ck})$  を実行し、 $\text{PK}$  を  $\mathcal{A}$  へ送る.

**Query Phase:**  $\mathcal{A}$  から  $\mathcal{S}$  への各オラクルクエリに対する  $\mathcal{S}$  の動作は以下の通りである.

**送信者用秘密鍵生成:**  $\mathcal{A}$  が属性集合  $S_s$  をクエリする.  $\mathcal{S}$  は  $S_s$  を  $\mathcal{CH}$  へ鍵生成オラクルクエリとして送り、 $\text{SK}_{S_s}$  を受け取り、 $\text{SK}_{S_s}$  を  $\mathcal{A}$  へ送る.

**受信者用秘密鍵生成:**  $\mathcal{A}$  が属性集合  $S_r$  をクエリする.  $\mathcal{S}$  は  $\text{CK.KeyGen}(\text{PK}_{ck}, \text{MK}_{ck}, S_r) \rightarrow \text{SK}_r$  を実行し、 $\text{SK}_r$  を  $\mathcal{A}$  へ送る.

**サインクリプション:**  $\mathcal{A}$  が平文  $m$ , 署名者アクセス構造  $\mathbb{A}_s$ , 復号者アクセス構造  $\mathbb{A}_d$ , 属性集合  $S_s \in \mathbb{A}_s$  をクエリする. 安全性証明の便宜上、この時点でリスト  $\mathcal{L}_{\text{SC},cs}$  の要素数を  $i-1$  として、 $\mathcal{A}$  から送られた先述のクエリをそれぞれ以下のように表記する.  $m_{\text{SC},cs}^{(i)} := m$ ,  $\mathbb{A}_s^{(i)} := \mathbb{A}_s$ ,  $\mathbb{A}_d^{(i)} := \mathbb{A}_d$ ,  $S_s^{(i)} := S_s$ .  $\mathcal{S}$  は  $\text{CK.Encap}(\text{PK}_{ck}, \mathbb{A}_d^{(i)}) := (K_{\text{SC},cs}^{(i)}, C_{K_{\text{SC},cs}}^{(i)})$  を実行し、 $(m_{\text{SC},cs}^{(i)} \| C_{K_{\text{SC},cs}}^{(i)}, S_s^{(i)}, \mathbb{A}_s^{(i)})$  を  $\mathcal{CH}$  へ署名生成オラクルクエリとして送り、 $\sigma_{\text{SC},cs}^{(i)}$  を受け取る.  $\mathcal{S}$  は

$D.\text{Encap}(K_{\text{SC},cs}^{(i)}, m_{\text{SC},cs}^{(i)} \parallel \sigma_{\text{SC},cs}^{(i)}) =: C_{D\text{SC},cs}^{(i)}, C_{\text{SC},cs}^{(i)} := (C_{K\text{SC},cs}^{(i)}, C_{D\text{SC},cs}^{(i)})$   
 を実行し,  $C_{\text{SC},cs}^{(i)}$  を  $\mathcal{A}$  へ送り, リスト  $\mathcal{L}_{\text{SC},cs} \leftarrow (m_{\text{SC},cs}^{(i)}, C_{\text{SC},cs}^{(i)}, \mathbb{A}_s^{(i)}_{\text{SC},cs}, \mathbb{A}_d^{(i)}_{\text{SC},cs}) = (m_{\text{SC},cs}^{(i)}, (C_{K\text{SC},cs}^{(i)}, C_{D\text{SC},cs}^{(i)}), \mathbb{A}_s^{(i)}_{\text{SC},cs}, \mathbb{A}_d^{(i)}_{\text{SC},cs})$  を追加する.

**アンサインクリプション:**  $\mathcal{A}$  はサインクリプテキスト  $C = (C_K, C_D)$ , 署名者アクセス構造  $\mathbb{A}_s$ , 属性集合  $S_r$  をクエリする.  $\mathcal{S}$  は  $\text{CK.KeyGen}(\text{PK}_{ck}, \text{MK}_{ck}, S_r) \rightarrow \text{SK}_r$ ,  $\text{CK.Decap}(\text{PK}_{ck}, C_K, \text{SK}_r) =: \alpha$  を実行する. もし,  $\alpha \neq \perp$  ならば,  $K := \alpha$ ,  $D.\text{Decap}(K, C_D) =: \beta$  を実行する. もし,  $\beta \neq \perp$  ならば,  $m \parallel \sigma = \beta$ ,  $\text{SS.Ver}(\text{PK}_{ss}, \sigma, m \parallel C_K, \mathbb{A}_s) =: \gamma$  を実行する. もし,  $\gamma = 1$  ならば,  $\mathcal{S}$  は  $\mathcal{A} \leftarrow m$  を返す. もし,  $\alpha = \perp$  または  $\beta = \perp$  または  $\gamma = 0$  ならば,  $\perp$  を返す.

**Forgery Phase:**  $\mathcal{A}$  は  $C^* = (C_K^*, C_D^*), \mathbb{A}_s^*, \mathbb{A}_d^*$  を出力する. 安全性証明の便宜上, 各変数の表記を以下のように変更する.  $C_{\text{Frg},cs}^* := C^*, C_{K\text{Frg},cs}^* := C_K^*, C_{D\text{Frg},cs}^* := C_D^*, \mathbb{A}_s^*_{\text{Frg},cs} := \mathbb{A}_s^*, \mathbb{A}_d^*_{\text{Frg},cs} := \mathbb{A}_d^*$ .  $\mathbb{A}_d^*_{\text{Frg},cs}$  の  $i$  番目の要素を  $S_r^{(i)}_{\text{Frg},cs}$  と表記する ( $i \in \{1, \dots, |\mathbb{A}_d^*_{\text{Frg},cs}|\}$ ).  $\mathcal{S}$  は,  $\text{CK.KeyGen}(\text{PK}_{ck}, \text{MK}_{ck}, S_r^{(i)}_{\text{Frg},cs}) =: \text{SK}_r^{(i)}_{\text{Frg},cs}$ ,  $\text{CK.Decap}(\text{PK}_{ck}, C_{K\text{Frg},cs}^*, \text{SK}_r^{(i)}_{\text{Frg},cs}) =: \alpha^{(i)}$  を実行する. もし,  $\alpha^{(i)} \neq \perp$  ならば,  $\mathcal{S}$  は  $K_{\text{Frg},cs}^{(i)} := \alpha^{(i)}$ ,  $D.\text{Decap}(K_{\text{Frg},cs}^{(i)}, C_{D\text{Frg},cs}^*) =: \beta^{(i)}$  を実行する. もし,  $\beta^{(i)} \neq \perp$  ならば,  $\mathcal{S}$  は  $m_{\text{Frg},cs}^{(i)} \parallel \sigma_{\text{Frg},cs}^{(i)} := \beta^{(i)}$  を実行し,  $\text{SS.Ver}(\text{PK}_{ss}, \sigma_{\text{Frg},cs}^{(i)}, m_{\text{Frg},cs}^{(i)} \parallel C_{K\text{Frg},cs}^*, \mathbb{A}_s^*_{\text{Frg},cs}) = 1$  ならば,  $\gamma_{\text{Frg},cs}^{(i)} = 1$  とする.  $\text{SS.Ver}(\text{PK}_{ss}, \sigma_{\text{Frg},cs}^{(i)}, m_{\text{Frg},cs}^{(i)} \parallel C_{K\text{Frg},cs}^*, \mathbb{A}_s^*_{\text{Frg},cs}) = 0$ , または  $\alpha^{(i)} = \perp$ , または  $\beta^{(i)} = \perp$  ならば,  $\gamma_{\text{Frg},cs}^{(i)} = 0$  とする. ここまでの一連の処理を全ての  $i \in \{1, \dots, |\mathbb{A}_d^*_{\text{Frg},cs}|\}$  について実行し終わったら,  $\mathcal{S}$  は  $\gamma_{\text{Frg},cs} := \gamma_{\text{Frg},cs}^{(1)} \times \gamma_{\text{Frg},cs}^{(2)} \times \dots \times \gamma_{\text{Frg},cs}^{(|\mathbb{A}_d^*_{\text{Frg},cs}|\)}$  を実行する.  $\mathcal{S}$  は,  $j \xleftarrow{\text{U}} \{1, \dots, |\mathbb{A}_d^*_{\text{Frg},cs}|\}$  を実行し,  $\mathcal{CH}$  に対して AP-sEUF-CMA ゲームの Forgery Phase で,  $(m_{\text{Frg},cs}^{(j)} \parallel C_{K\text{Frg},cs}^*, \sigma_{\text{Frg},cs}^{(j)}, \mathbb{A}_s^*_{\text{Frg},cs})$  を出力する.

$\mathcal{S}$  は  $\mathcal{A}$  に対して  $\Pi_{\text{CS}}$  に関する AP-sEUF-CMA ゲームの挑戦者としての応答を完璧にシミュレートできている事はほぼ自明であるので, 詳細な説明は割愛する.

$\mathcal{S}$  は  $\mathcal{A}$  が  $\Pi_{\text{CS}}$  に関する AP-sEUF-CMA ゲームにおける禁止されたオラクルクエリを一度も行わなければ,  $\Pi_{\text{SS}}$  に関する AP-sEUF-CMA ゲームにおいて禁止されたオラクルクエリを行うことはない.

従って,  $\Pi_{\text{CS}}$  に関する AP-sEUF-CMA 安全性ゲームにおける  $\mathcal{A}$  の優位性はこれま

で定義した変数・記号を使って次式で定義できる.

$$\begin{aligned}
\text{Adv}_{\Pi_{CS}, \mathcal{A}}^{\text{AP-sEUF-CMA}} &= \Pr[ [\text{Disclose}_{CS}(\text{PK}, C) = \text{Disclose}_{CK}(\text{PK}_{ck}, C_K^*) = \mathbb{A}_d^*] \wedge [\gamma_{\text{Frg}, cs} = 1] \\
&\quad \wedge [m_{\text{Frg}, cs}^{(1)} = \dots = m_{\text{Frg}, cs}^{(\mathbb{A}_{d\text{Frg}, cs}^*)} =: m_{\text{Frg}, cs}^*] \\
&\quad \wedge [(m_{\text{Frg}, cs}^*, (C_{K\text{Frg}, cs}^*, C_{D\text{Frg}, cs}^*), \mathbb{A}_{s\text{Frg}, cs}^*, \mathbb{A}_{d\text{Frg}, cs}^*) \notin \mathcal{L}_{\text{SC}, cs}] ]
\end{aligned} \tag{5.19}$$

ここで,  $\mathcal{CH}$  の動作を考える. 具体的には, Query Phase の署名生成オラクルでの動作と, Forgery Phase での動作について考える.

まず, Query Phase の署名生成オラクルにおいて,  $\mathcal{CH}$  は,  $|\mathcal{L}_{\text{SC}, cs}| = i - 1$  である時点で,  $\mathcal{S}$  より,  $(m_{\text{SC}, cs}^{(i)} \| C_{K\text{SC}, cs}^{(i)}, S_{s\text{SC}, cs}^{(i)}, \mathbb{A}_{s\text{SC}, cs}^{(i)})$  をクエリされる.  $\mathcal{S}$  は,  $\mathcal{A}$  がサインクリプションオラクルにクエリした時に署名生成オラクルへのクエリを行う以外に, 署名生成オラクルへのクエリを行うことはないと仮定できるので,  $\mathcal{S}$  が署名生成オラクルへクエリを行う時点では常に  $|\mathcal{L}_{\text{SC}, cs}| = |\mathcal{L}_{\text{Sig}, ss}|$  が成立する. 従って,  $\mathcal{S}$  が  $(m_{\text{SC}, cs}^{(i)} \| C_{K\text{SC}, cs}^{(i)}, S_{s\text{SC}, cs}^{(i)}, \mathbb{A}_{s\text{SC}, cs}^{(i)})$  をクエリしてきた時,  $|\mathcal{L}_{\text{Sig}, ss}| = i - 1$  である.  $k \in \{1, \dots, i - 1\}$  について,  $\mathcal{L}_{\text{SC}, cs}$  の  $k$  番目の要素と,  $\mathcal{L}_{\text{Sig}, ss}$  の  $k$  番目の要素は, この時点で一対一で対応している.  $\mathcal{CH}$  はこのオラクルクエリに対して次のように動作する. まず,  $\text{SS.KeyGen}(\text{PK}_{ss}, \text{MK}_{ss}, S_{s\text{SC}, cs}^{(i)}) =: \text{SK}_s^{(i)} \text{Sig}, ss$  を実行する. 次に,  $\text{SS.Sig}(\text{PK}_{ss}, m_{\text{SC}, cs}^{(i)} \| C_{K\text{SC}, cs}^{(i)}, \text{SK}_s^{(i)} \text{Sig}, ss, \mathbb{A}_{s\text{SC}, cs}^{(i)}) =: \sigma_{\text{Sig}, ss}^{(i)}$  を実行する. そして,  $\sigma_{\text{Sig}, ss}^{(i)}$  を署名として  $\mathcal{S}$  へ送る. ここで,  $\sigma_{\text{Sig}, ss}^{(i)} = \sigma_{\text{Sig}, cs}^{(i)}$  であることは自明である. 最後に,  $m_{\text{Sig}, ss}^{(i)} := m_{\text{SC}, cs}^{(i)} \| C_{K\text{SC}, cs}^{(i)}, \mathbb{A}_{s\text{Sig}, ss}^{(i)} := \mathbb{A}_{s\text{SC}, cs}^{(i)}$  を実行し, リスト  $\mathcal{L}_{\text{Sig}, ss}$  へ,  $(m_{\text{Sig}, ss}^{(i)}, \sigma_{\text{Sig}, ss}^{(i)}, \mathbb{A}_{s\text{Sig}, ss}^{(i)})$  を追加する.

次に, Forgery Phase における,  $\mathcal{CH}$  の動作について考える.  $\mathcal{CH}$  は  $\mathcal{S}$  より,  $(m_{\text{Frg}, cs}^{(j)} \| C_{K\text{Frg}, cs}^*, \sigma_{\text{Frg}, cs}^{(j)}, \mathbb{A}_{s\text{Frg}, cs}^*) (j \in \{1, \dots, |\mathbb{A}_{d\text{Frg}, cs}^*|\})$  を出力される.  $\mathcal{CH}$  は署名の正当性の検証のため,  $\text{SS.Ver}(\text{PK}_{ss}, \sigma_{\text{Frg}, cs}^{(j)}, m_{\text{Frg}, cs}^{(j)} \| C_{K\text{Frg}, cs}^*, \mathbb{A}_{s\text{Frg}, cs}^*) =: \gamma_{\text{Frg}, ss} = 1 / 0$  を実行する. ここで,  $\text{SS.Ver}$  は確定的アルゴリズムであるため, 入力変数が同一であれば, 出力は必ず同一になるから, 次の等式が成り立つ.

$$\gamma_{\text{Frg}, ss} = \gamma_{\text{Frg}, cs}^{(j)} \tag{5.20}$$

これまでに定義した変数, 記号を使って, SP-ABS 方式  $\Pi_{SS}$  に関する AP-sEUF-CMA 安全性ゲームにおける  $\mathcal{S}$  の優位性は次式で定義できる.

$$\text{Adv}_{\Pi_{SS}, \mathcal{S}}^{\text{AP-sEUF-CMA}} = \Pr[ [\gamma_{\text{Frg}, ss} = 1] \wedge [(m_{\text{Frg}, cs}^{(j)} \| C_{K\text{Frg}, cs}^*, \sigma_{\text{Frg}, cs}^{(j)}, \mathbb{A}_{s\text{Frg}, cs}^*) \notin \mathcal{L}_{\text{Sig}, ss}] ] \tag{5.21}$$

ここで、安全性証明の便宜上、事象  $P_1, P_2, P_3, P_4, Q_1, Q_2$  を次の通りに定義する。

$$P_1 = [\text{Disclose}_{\text{CS}}(\text{PK}, C) = \text{Disclose}_{\text{CK}}(\text{PK}_{ck}, C_K^*) = \mathbb{A}_d^*] \quad (5.22)$$

$$P_2 = [\gamma_{\text{Frg},cs} = 1] \quad (5.23)$$

$$P_3 = [m_{\text{Frg},cs}^{(1)} = \cdots = m_{\text{Frg},cs}^{(|\mathbb{A}_d^*_{\text{Frg},cs}|)} =: m_{\text{Frg},cs}^*] \quad (5.24)$$

$$P_4 = [(m_{\text{Frg},cs}^*, (C_{K\text{Frg},cs}^*, C_{D\text{Frg},cs}^*), \mathbb{A}_{s\text{Frg},cs}^*, \mathbb{A}_{d\text{Frg},cs}^*) \notin \mathcal{L}_{\text{SC},cs}] \quad (5.25)$$

$$Q_1 = [\gamma_{\text{Frg},ss} = 1] \quad (5.26)$$

$$Q_2 = [(m_{\text{Frg},cs}^{(j)} \| C_{K\text{Frg},cs}^*, \sigma_{\text{Frg},cs}^{(j)}, \mathbb{A}_{s\text{Frg},cs}^*) \notin \mathcal{L}_{\text{Sig},ss}] \quad (5.27)$$

(以降で証明する) 補題 5.2.2.1 と補題 5.2.2.2 より、事象  $P := P_1 \wedge P_2 \wedge P_3 \wedge P_4$  が生じた場合、事象  $Q_1$  と事象  $Q_2$  はどちらも生起する。従って、次の不等式が成立する。

$$\text{Adv}_{\Pi_{\text{SS}},S}^{\text{AP-sEUF-CMA}} \geq \text{Adv}_{\Pi_{\text{CS}},\mathcal{A}}^{\text{AP-sEUF-CMA}} \quad (5.28)$$

$\text{Adv}_{\Pi_{\text{CS}},\mathcal{A}}^{\text{AP-sEUF-CMA}}$  が無視できなくなるような  $\mathcal{A}$  が存在すると仮定すると、式 (5.28) より、 $\text{Adv}_{\Pi_{\text{SS}},S}^{\text{AP-sEUF-CMA}}$  は無視できなくなり、これは  $\Pi_{\text{SS}}$  が AP-sEUF-CMA 安全であるという事実と矛盾するので、その仮定は誤りである。従って、 $\Pi_{\text{SS}}$  が AP-sEUF-CMA 安全ならば、いかなる PPTA  $\mathcal{A}$  に対しても、 $\text{Adv}_{\Pi_{\text{CS}},\mathcal{A}}^{\text{AP-sEUF-CMA}}$  は無視できるほど小さい値になる。ゆえに、補題 5.2.2 は成立する。  $\square$

**補題 5.2.2.1.** 事象  $P = P_1 \wedge P_2 \wedge P_3 \wedge P_4$  が生じた場合、事象  $Q_1$  は生起する。つまり、 $\Pr[Q_1|P] = 1$  が成立する。

**補題 5.2.2.2.** 事象  $P = P_1 \wedge P_2 \wedge P_3 \wedge P_4$  が生じた場合、事象  $Q_2$  は生起する。つまり、 $\Pr[Q_2|P] = 1$  が成立する。

以降では、補題 5.2.2.1、補題 5.2.2.2 の証明を示す。

**補題 5.2.2.1 の証明** 事象  $P_2$  は生起することが仮定されているため、

$$\gamma_{\text{Frg},cs} = 1 \quad (5.29)$$

$\gamma_{\text{Frg},cs}$  の定義より、

$$\gamma_{\text{Frg},cs} = \gamma_{\text{Frg},cs}^{(1)} \times \gamma_{\text{Frg},cs}^{(2)} \times \cdots \times \gamma_{\text{Frg},cs}^{(|\mathbb{A}_d^*_{\text{Frg},cs}|)} \quad (5.30)$$

式 (5.29)、式 (5.30) より、 $i \in \{1, \dots, |\mathbb{A}_d^*_{\text{Frg},cs}|\}$  を満たす全ての  $i$  について、次の等式が成立。

$$\gamma_{\text{Frg},cs}^{(i)} = 1 \quad (5.31)$$

式 (5.31)、式 (5.20) より、次の等式が成立。

$$\gamma_{\text{Frg},ss} = \gamma_{\text{Frg},cs}^{(j)} = 1 \quad (5.32)$$

従って、事象  $P$  が生じた場合、事象  $Q_1$  は生起する。ゆえに、補題 5.2.2.1 は成立する。  $\square$

**補題 5.2.2.2 の証明** 事象  $P$  が生じた場合に事象  $Q_2$  は生じないと仮定すると矛盾が生じることを示す. その仮定の下では, 次の等式を満たすリスト  $\mathcal{L}_{\text{Sig},s}$  内の  $k$  番目の要素  $(m_{\text{SC},cs}^{(k)} \| C_{K \text{ SC},cs}^{(k)}, \sigma_{\text{SC},cs}^{(k)}, \mathbb{A}_{s \text{ SC},cs}^{(k)})$  が存在する.

$$(m_{\text{SC},cs}^{(k)} \| C_{K \text{ SC},cs}^{(k)}, \sigma_{\text{SC},cs}^{(k)}, \mathbb{A}_{s \text{ SC},cs}^{(k)}) = (m_{\text{Frg},cs}^{(j)} \| C_{K \text{ Frg},cs}^*, \sigma_{\text{Frg},cs}^{(j)}, \mathbb{A}_{s \text{ Frg},cs}^*) \quad (5.33)$$

式 (5.33) より以下の 4 つの等式が成立する.

$$m_{\text{SC},cs}^{(k)} = m_{\text{Frg},cs}^{(j)} \quad (5.34)$$

$$C_{K \text{ SC},cs}^{(k)} = C_{K \text{ Frg},cs}^* \quad (5.35)$$

$$\sigma_{\text{SC},cs}^{(k)} = \sigma_{\text{Frg},cs}^{(j)} \quad (5.36)$$

$$\mathbb{A}_{s \text{ SC},cs}^{(k)} = \mathbb{A}_{s \text{ Frg},cs}^* \quad (5.37)$$

$C_{K \text{ SC},cs}^{(k)}$  は  $\text{CK.Encap}(\text{PK}_{ck}, \mathbb{A}_d^{(k)} \text{ SC},cs)$  の実行により生成されている正当な鍵暗号文である事実と,  $\Pi_{\text{CK}}$  は復号者アクセス構造開示性を満たし, アルゴリズム  $\text{Disclose}_{\text{CK}}$  が存在することから, 以下の等式が成立する.

$$\Pr[\text{Disclose}_{\text{CK}}(C_{K \text{ SC},cs}^{(k)}) = \mathbb{A}_d^{(k)} \text{ SC},cs] = 1 \quad (5.38)$$

一方で, 事象  $P_1$  は生起すると仮定されているので, 以下の等式が成立する.

$$\Pr[\text{Disclose}_{\text{CK}}(C_{K \text{ Frg},cs}^*) = \mathbb{A}_{s \text{ Frg},cs}^*] = 1 \quad (5.39)$$

式 (5.35), (5.38), (5.39) より, 以下の等式が成立する.

$$\mathbb{A}_d^{(k)} \text{ SC},cs = \mathbb{A}_{d \text{ Frg},cs}^* \quad (5.40)$$

$C_{K \text{ SC},cs}^{(k)}$  は,  $\mathcal{S}$  が  $\text{CK.Encap}(\text{PK}_{ck}, \mathbb{A}_d^{(k)} \text{ SC},cs) \rightarrow (K_{\text{SC},cs}^{(k)}, C_{K \text{ SC},cs}^{(k)})$  を実行して生成したものであるので,  $i \in \{1, \dots, |\mathbb{A}_d^{(k)} \text{ SC},cs|\}$ ,  $S_r^{(i)} \text{ SC},cs \in \mathbb{A}_d^{(k)} \text{ SC},cs$ ,  $\text{CK.KeyGen}(\text{PK}_{ck}, \text{MK}_{ck}, S_r^{(i)} \text{ SC},cs) =: \text{SK}_r^{(i)} \text{ SC},cs$  を定義すると, CP-ABKEM 方式  $\Pi_{\text{CK}}$  の正当性より, 全ての  $i \in \{1, \dots, |\mathbb{A}_d^{(k)} \text{ SC},cs|\}$  について, 以下の等式が成立する.

$$\Pr[\text{CK.Decap}(\text{PK}_{ck}, C_{K \text{ SC},cs}^{(k)}, \text{SK}_r^{(i)} \text{ SC},cs) = K_{\text{SC},cs}^{(k)}] = 1 \quad (5.41)$$

式 (5.35), (5.40), (5.41) より, 全ての  $i \in \{1, \dots, |\mathbb{A}_{d \text{ Frg},cs}^*|\}$  について,  $K_{\text{Frg},cs}^*$  が存在し, 以下の等式が成立する.

$$K_{\text{Frg},cs}^{(i)} = K_{\text{SC},cs}^{(k)} = K_{\text{Frg},cs}^* \quad (5.42)$$

$i \in \{1, \dots, |\mathbb{A}_{d \text{ Frg},cs}^*|\}$  について,  $m_{\text{Frg},cs}^{(i)}$  と  $\sigma_{\text{Frg},cs}^{(i)}$  は,  $\mathcal{S}$  が  $\text{D.Decap}(K_{\text{Frg},cs}^{(i)}, C_{D \text{ Frg},cs}^*)$  を実行して生成したものである事実と, 等式 (5.42) より, 全ての  $i \in \{1, \dots, |\mathbb{A}_{d \text{ Frg},cs}^*|\}$  について, 先述の  $\text{D.Decap}$  アルゴリズムへの入力変数は同じであり,  $\text{D.Decap}$  は確定的

アルゴリズムであるので入力変数が同じであれば出力は同じになることから、全ての  $i \in \{1, \dots, |\mathbb{A}_{d\text{Frg},cs}^*|\}$  について、 $m_{\text{Frg},cs}^*$  と  $\sigma_{\text{Frg},cs}^*$  が存在し、以下の等式が成立する.

$$m_{\text{Frg},cs}^{(i)} \parallel \sigma_{\text{Frg},cs}^{(i)} = m_{\text{Frg},cs}^* \parallel \sigma_{\text{Frg},cs}^* \quad (5.43)$$

式 (5.34), (5.36), (5.43) より、以下の等式が成立する.

$$m_{\text{SC},cs}^{(k)} = m_{\text{Frg},cs}^{(j)} = m_{\text{Frg},cs}^* \quad (5.44)$$

$$\sigma_{\text{SC},cs}^{(k)} = \sigma_{\text{Frg},cs}^{(j)} = \sigma_{\text{Frg},cs}^* \quad (5.45)$$

$i \in \{1, \dots, |\mathbb{A}_{d\text{Frg},cs}^*|\}$  について、 $m_{\text{Frg},cs}^{(i)}$  と  $\sigma_{\text{Frg},cs}^{(i)}$  は、 $\mathcal{S}$  が  $\text{D.Decap}(K_{\text{Frg},cs}^{(i)}, C_{D\text{Frg},cs}^*)$  を実行して生成したものである事実と、式 (5.42), (5.43) より、以下の等式が成立する.

$$\Pr[\text{D.Decap}(K_{\text{Frg},cs}^*, C_{D\text{Frg},cs}^*) = m_{\text{Frg},cs}^* \parallel \sigma_{\text{Frg},cs}^*] = 1 \quad (5.46)$$

$C_{D\text{SC},cs}^{(k)}$  は、 $\mathcal{S}$  が  $\text{D.Encap}(K_{\text{SC},cs}^{(k)}, m_{\text{SC},cs}^{(k)} \parallel \sigma_{\text{SC},cs}^{(k)})$  を実行して生成したものである事実と、DEM 方式  $\Pi_D$  の正当性より、以下の等式が成立する.

$$\Pr[\text{D.Decap}(K_{\text{SC},cs}^{(k)}, C_{D\text{SC},cs}^{(k)}) = m_{\text{SC},cs}^{(k)} \parallel \sigma_{\text{SC},cs}^{(k)}] = 1 \quad (5.47)$$

式 (5.42), (5.44), (5.45), (5.46), (5.47),  $\Pi_D$  の一対一対応性より、以下の等式が成立する.

$$C_{D\text{SC},cs}^{(k)} = C_{D\text{Frg},cs}^* \quad (5.48)$$

式 (5.35), (5.37), (5.40), (5.44), (5.48) より、以下の等式が成立する.

$$(m_{\text{SC},cs}^{(k)}, (C_{K\text{SC},cs}^{(k)}, C_{D\text{SC},cs}^{(k)}), \mathbb{A}_s^{(k)}, \mathbb{A}_d^{(k)}) = (m_{\text{Frg},cs}^*, (C_{K\text{Frg},cs}^*, C_{D\text{Frg},cs}^*), \mathbb{A}_s^*, \mathbb{A}_d^*) \quad (5.49)$$

等式 (5.49) は事象  $P$  が生じた場合という条件に対する矛盾を表しているので、事象  $Q_2$  が生じないという仮定が誤りである. ゆえに、事象  $P$  が生じた場合、事象  $Q_2$  は必ず生起する. ゆえに、補題 5.2.2.2 は成立する.  $\square$

**定理 5.3 の証明** 任意の  $k$ , 任意の  $\mathcal{U}_s$ , 任意の  $\mathcal{U}_r$  に対し、 $(\text{PK}_{ss}, \text{MK}_{ss}) \leftarrow \text{SS.Setup}(1^k, \mathcal{U}_s)$  を実行し、 $(\text{PK}_{ck}, \text{MK}_{ck}) \leftarrow \text{CK.Setup}(1^k, \mathcal{U}_s)$  を実行する. さらに、任意の  $S_s \in (2^{\mathcal{U}_s} - \{\emptyset\})$ , 任意の  $S'_s \in (2^{\mathcal{U}_s} - \{\emptyset\})$  に対し、 $\text{SK}_s \leftarrow \text{SS.KeyGen}(\text{PK}_{ss}, \text{MK}_{ss}, S_s)$ ,  $\text{SK}'_s \leftarrow \text{SS.KeyGen}(\text{PK}_{ss}, \text{MK}_{ss}, S'_s)$  を実行する. そして、任意の  $m \in \mathcal{M}$ , 任意の  $\mathbb{A}_s$  (s.t.  $S_s \in \mathbb{A}_s \wedge S'_s \in \mathbb{A}_s$ ), 任意の  $\mathbb{A}_d$  に対して、以下の一連の処理を実行する.  $(K, C_K) \leftarrow \text{CK.Encap}(\text{PK}_{ck}, \mathbb{A}_d)$ ,  $\sigma \leftarrow \text{SS.Sig}(\text{PK}_{ss}, m \parallel C_K, \text{SK}_s, \mathbb{A}_s)$ ,  $\sigma' \leftarrow \text{SS.Sig}(\text{PK}_{ss}, m \parallel C_K, \text{SK}'_s, \mathbb{A}_s)$ ,  $C_D \leftarrow \text{D.Encap}(K, m \parallel \sigma)$ ,  $C'_D \leftarrow \text{D.Encap}(K, m \parallel \sigma')$ ,  $C := (C_K, C_D)$ ,  $C' := (C_K, C'_D)$ .

ここで、 $\text{SP-ABS}\Pi_{\text{SS}}$  は完全匿名だから、 $\sigma$  の確率分布と、 $\sigma'$  の確率分布は同一である.

そして、 $\sigma$  の確率分布と  $\sigma'$  の確率分布が同一であれば、 $m \parallel \sigma$  の確率分布と  $m \parallel \sigma'$  の確率分布は同一である.

さらに,  $m\|\sigma$  の確率分布と  $m\|\sigma'$  の確率分布が同一であれば,  $C_D$  の確率分布と  $C'_D$  の確率分布は同一である.

最後に,  $C_D$  の確率分布と  $C'_D$  の確率分布が同一であれば,  $C$  の確率分布と  $C'$  の確率分布は同一である.

従って, CP-ABSC の一般的構成  $\Pi_{CS}$  は, 完全匿名である. ゆえに, 定理 5.3 が成立する. □

# Chapter 6 鍵ポリシー型属性ベース鍵カプセル化メカニズムの一般的構成

## 6.1 本章の概要

本章に示す成果は、鍵ポリシー型属性ベース暗号 (KP-ABE) を構成要素とする、鍵ポリシー型属性ベース鍵カプセル化メカニズム (KP-ABKEM) の一般的構成法とその安全性証明である。具体的には、AA-IND-CCA 安全、かつ復号者属性集合 開示的な KP-ABE から、AA-IND-CCA 安全、かつ復号者属性集合開示的な KP-ABKEM を一般的に構成できることを証明した。

なお、本成果 (鍵ポリシー型属性ベース鍵カプセル化メカニズムの一般的構成) の意義については、8.1.1 項で説明する。

また、本章の構成については以下の通りである。6.2 節で一般的構成法の説明を行う。6.3 節で安全性証明の詳述を行う。

## 6.2 提案する一般的構成法

構成要素として、KP-ABE 方式  $\Pi_{KE} : (\text{KE.Setup}, \text{KE.KeyGen}, \text{KE.Enc}, \text{KE.Dec})$  を用いた、KP-ABKEM 方式  $\Pi_{KK} : (\text{KK.Setup}, \text{KK.KeyGen}, \text{KK.Encap}, \text{KK.Decap})$  の一般的構成法を図 6.1 に示す。

KK.Setup( $1^k, \mathcal{U}$ ) : Return $(\text{PK}, \text{MK}) \leftarrow \text{KE.Setup}(1^k, \mathcal{U})$ .
KK.KeyGen( $\text{PK}, \text{MK}, \mathbb{A}$ ) : Return $\text{SK} \leftarrow \text{KE.KeyGen}(\text{PK}, \text{MK}, \mathbb{A})$ .
KK.Encap( $\text{PK}, S_d$ ) : $K \xleftarrow{\mathcal{U}} \mathcal{K}; C_K \leftarrow \text{KE.Enc}(\text{PK}, K, S_d);$ Return $(K, C_K)$ .
KK.Decap( $\text{PK}, C_K, \text{SK}$ ) : $\alpha := \text{KE.Dec}(\text{PK}, C_K, \text{SK}_r);$ If $\alpha = \perp$ , then return $\perp$ . Else return $K := \alpha$ .

図 6.1: KP-ABKEM 方式の一般的構成法  $\Pi_{KK}$

### 6.3 安全性証明

図 6.1 の KP-ABKEM 方式の一般的構成法  $\Pi_{\text{KK}}$  に関して, 定理 6.1, 定理 6.2 が成立する.

**定理 6.1.** KP-ABE 方式  $\Pi_{\text{KE}}$  が AA-IND-CCA 安全であれば, 図 6.1 の KP-ABKEM 方式  $\Pi_{\text{KK}}$  は AA-IND-CCA 安全である.

**定理 6.2.** KP-ABE 方式  $\Pi_{\text{KE}}$  が復号者アクセス構造開示性を満たすならば, 図 6.1 の KP-ABKEM 方式  $\Pi_{\text{KK}}$  は復号者属性集合開示性を満たす.

**定理 6.1 の証明** 証明の手順は補題 4.1 の証明と同様である. 冗長的にならないよう, 証明の詳細な記述を所々で省略する.

$\text{Game}_0, \text{Game}_1$  の安全性ゲームを以下のように定義する.

**Game<sub>0</sub>:** 攻撃者と挑戦者が行う,  $\Pi_{\text{KK}}$  に関する AA-IND-CCA 安全性ゲーム.

**Game<sub>1</sub>:**  $\text{Game}_0$  に以下の変更を加えたゲームとする.

- **Challenge フェーズ**において,  $\mathcal{CH}$  はチャレンジビットの決定をチャレンジ鍵暗号文  $C_K^*$  の計算の前に行う. 更に, 鍵暗号文  $C_K^*$  は鍵  $K_1$  ではなく, 鍵  $K_b$  を暗号化して生成する. また,  $\mathcal{A}$  に対してチャレンジ鍵としてチャレンジビット  $b$  が  $b=0$  であっても  $b=1$  であっても  $K_1$  を返す. 具体的には,  $\mathcal{CH}$  は  $\mathcal{A}$  から  $S_d^*$  を受け取り,  $K_0, K_1 \xleftarrow{\text{U}} \mathcal{K}, b \xleftarrow{\text{U}} \{0, 1\}, \text{KE.Enc}(\text{PK}, K_b, S_d^*) \rightarrow C_K^*$  を実行し, 最終的に  $(K_1, C_K^*)$  を  $\mathcal{A}$  へ返す.

$\text{Game}_i$  の Guess フェーズで攻撃者  $\mathcal{A}$  が正しい推測ビット  $b' = b$  を出力する事象を  $W_i$  と表記する.  $\text{Game}_0$  における攻撃者  $\mathcal{A}$  の利得の定義式より,

$$\text{Adv}_{\Pi_{\text{KK}}, \mathcal{A}}^{\text{AA-IND-CCA}} = |\Pr[W_0] - \frac{1}{2}| \leq |\Pr[W_0] - \Pr[W_1]| + |\Pr[W_1] - \frac{1}{2}| \quad (6.1)$$

不等式 (6.1) と以下で証明する補題 6.1.1 及び補題 6.1.2 より,  $\Pi_{\text{KE}}$  が AA-IND-CCA 安全であるならば, いかなる PPTA 攻撃者  $\mathcal{A}$  に対しても,  $\text{Adv}_{\Pi_{\text{KK}}, \mathcal{A}}^{\text{AA-IND-CCA}}$  は無視できるほど小さい. ゆえに, 定理 6.1 が成立する.  $\square$

**補題 6.1.1.** いかなる PPTA 攻撃者  $\mathcal{A}$  に対しても,  $|\Pr[W_0] - \Pr[W_1]|$  は無視できるほど小さい値になる.

**補題 6.1.2.** KP-ABE 方式  $\Pi_{\text{KE}}$  が AA-IND-CCA 安全ならば, いかなる PPTA 攻撃者  $\mathcal{A}$  に対しても,  $|\Pr[W_1] - \frac{1}{2}|$  は無視できるほど小さい値になる.

**補題 6.1.1 の証明** 証明の手順は補題 4.1.1 の証明と同様である。冗長的にならないよう、証明の詳細な記述を所々で省略する。

$\text{Game}_0$  における PPTA 攻撃者を  $\mathcal{A}$  とし、挑戦者を  $\mathcal{CH}$  とする。  $\text{Game}_0$  において  $\mathcal{A}$  と  $\mathcal{CH}$  は以下の通りに動作する。

**Setup Phase:**  $\mathcal{CH}$  は  $\text{KE.Setup}$  を実行し、システム公開鍵  $\text{PK}$  を  $\mathcal{A}$  に渡す。

**Query Phase 1:**  $\mathcal{A}$  は以下の各オラクルに対して、クエリを任意回数発行できる。

**秘密鍵生成:**  $\mathcal{A}$  はアクセス構造  $\mathbb{A}$  を  $\mathcal{CH}$  へ送る。  $\mathcal{CH}$  は  $\text{CE.KeyGen}(\text{PK}, \text{MK}, \mathbb{A}) \rightarrow \text{SK}$  を計算し  $\text{SK}$  を  $\mathcal{A}$  へ送る。

**鍵復号:**  $\mathcal{A}$  は鍵暗号文  $C_K$ , アクセス構造  $\mathbb{A}$  を  $\mathcal{CH}$  へ送る。  $\mathcal{CH}$  は  $\text{KE.KeyGen}(\text{PK}, \text{MK}, \mathbb{A}) \rightarrow \text{SK}$  で秘密鍵を生成し、  $\text{KE.Dec}(\text{PK}, C_K, \text{SK}) \rightarrow K / \perp$  を実行し最後の出力結果を  $\mathcal{A}$  へ送る。

**Challenge Phase:**  $\mathcal{A}$  はターゲット復号者属性集合  $S_d^*$  を  $\mathcal{CH}$  へ送る。  $\mathcal{CH}$  は  $K_0, K_1 \xleftarrow{\mathcal{U}} \mathcal{K}, \text{KE.Enc}(\text{PK}, K_1, S_d^*) \rightarrow C_K^*, b \xleftarrow{\mathcal{U}} \{0, 1\}$  を実行し、  $(K_b, C_K^*)$  を  $\mathcal{A}$  に渡す。

**Query Phase 2:**  $\mathcal{A}$  は以下の各オラクルに対して、クエリを任意回数発行できる。

**秘密鍵生成:** Query Phase 1 と同じ。

**鍵復号:** Query Phase 1 と同じ。

**Guess Phase:**  $\mathcal{A}$  はチャレンジビット  $b$  に対する推測として、  $b' \in \{0, 1\}$  を出力。

同様に、  $\text{Game}_1$  における PPTA 攻撃者を  $\mathcal{A}$  とし、挑戦者を  $\mathcal{CH}$  とする。  $\text{Game}_1$  において  $\mathcal{A}$  と  $\mathcal{CH}$  は以下の通りに動作する。

**Setup Phase:**  $\mathcal{CH}$  は  $\text{KE.Setup}$  を実行し、システム公開鍵  $\text{PK}$  を  $\mathcal{A}$  に渡す。

**Query Phase 1:**  $\mathcal{A}$  は以下の各オラクルに対して、クエリを任意回数発行できる。

**秘密鍵生成:**  $\mathcal{A}$  はアクセス構造  $\mathbb{A}$  を  $\mathcal{CH}$  へ送る。  $\mathcal{CH}$  は  $\text{KE.KeyGen}(\text{PK}, \text{MK}, \mathbb{A}) \rightarrow \text{SK}$  を計算し  $\text{SK}$  を  $\mathcal{A}$  へ送る。

**鍵復号:**  $\mathcal{A}$  は鍵暗号文  $C_K$ , アクセス構造  $\mathbb{A}$  を  $\mathcal{CH}$  へ送る。  $\mathcal{CH}$  は  $\text{KE.KeyGen}(\text{PK}, \text{MK}, \mathbb{A}) \rightarrow \text{SK}$  で秘密鍵を生成し、  $\text{KE.Dec}(\text{PK}, C_K, \text{SK}) \rightarrow K / \perp$  を実行し最後の出力結果を  $\mathcal{A}$  へ送る。

**Challenge Phase:**  $\mathcal{A}$  はターゲット復号者属性集合  $S_d^*$  を  $\mathcal{CH}$  へ送る.  $\mathcal{CH}$  は  $K_0, K_1 \xleftarrow{\mathcal{U}} \mathcal{K}, b \xleftarrow{\mathcal{U}} \{0, 1\}, \text{KE.Enc}(\text{PK}, K_b, S_d^*) \rightarrow C_K^*$  を実行し,  $(K_1, C_K^*)$  を  $\mathcal{A}$  に渡す.

**Query Phase 2:**  $\mathcal{A}$  は以下の各オラクルに対して, クエリを任意回数発行できる.

**秘密鍵生成:** Query Phase 1 と同じ.

**鍵復号:** Query Phase 1 と同じ.

**Guess Phase:**  $\mathcal{A}$  はチャレンジビット  $b$  に対する推測として,  $b'$  を出力.

補題 4.1.1 の証明と同様に, 合計 10 個の変数  $K_{0,G_0}, K_{1,G_0}, C_{K,G_0}^*, b_{G_0}, b'_{G_0}, K_{0,G_1}, K_{1,G_1}, C_{K,G_1}^*, b_{G_1}, b'_{G_1}$  を定義する.

そして, 補題 4.1.1 の証明と同様に, 以下に示す 2 つの等式と 1 つの不等式 ((6.2), (6.3), (6.4)) が成立する.

$$\Pr[W_0] = \Pr[b_{G_0} = b'_{G_0}] = \frac{1}{2}(\Pr[b'_{G_0} = 0|b_{G_0} = 0] + \Pr[b'_{G_0} = 1|b_{G_0} = 1]) \quad (6.2)$$

$$\Pr[W_1] = \Pr[b_{G_1} = b'_{G_1}] = \frac{1}{2}(\Pr[b'_{G_1} = 0|b_{G_1} = 0] + \Pr[b'_{G_1} = 1|b_{G_1} = 1]) \quad (6.3)$$

$$\begin{aligned} |\Pr[W_0] - \Pr[W_1]| &\leq \frac{1}{2}(|\Pr[b'_{G_0} = 0|b_{G_0} = 0] - \Pr[b'_{G_1} = 0|b_{G_1} = 0]| \\ &\quad + |\Pr[b'_{G_0} = 1|b_{G_0} = 1] - \Pr[b'_{G_1} = 1|b_{G_1} = 1]|) \end{aligned} \quad (6.4)$$

式(6.4)と, 以下に示す補題 6.1.1.1 と補題 6.1.1.2 により, いかなる PPTA 攻撃者に対しても,  $|\Pr[W_0] - \Pr[W_1]|$  は無視できるほど小さい値になる. ゆえに, 補題 6.1.1 が成立する.  $\square$

**補題 6.1.1.1.** いかなる PPTA 攻撃者に対しても,  $|\Pr[b'_{G_0} = 1|b_{G_0} = 1] - \Pr[b'_{G_1} = 1|b_{G_1} = 1]|$  は無視できるほど小さい値になる.

**補題 6.1.1.2.** いかなる PPTA 攻撃者に対しても,  $|\Pr[b'_{G_0} = 0|b_{G_0} = 0] - \Pr[b'_{G_1} = 0|b_{G_1} = 0]|$  は無視できるほど小さい値になる.

補題 6.1.1.1 (resp. 補題 6.1.1.2) は, 補題 4.1.1.1 (resp. 補題 4.1.1.2) の証明の一か所のみを変更したもの (具体的には, 補題 4.1.1.1 (resp. 補題 4.1.1.2) の証明の 3 段落目にある “ $\text{CE.Enc}(\text{PK}, \cdot, A_d^*)$ ” を “ $\text{KE.Enc}(\text{PK}, \cdot, S_d^*)$ ” へ変更したもの) で証明可能であるため, 補題 6.1.1.1 (resp. 補題 6.1.1.2) の証明の詳細な記述は割愛する.

**補題 6.1.2 の証明** 証明の手順は補題 4.1.2 の証明と同様である．冗長的にならないよう，証明の詳細な記述を所々で省略する．

PPTA 攻撃者  $\mathcal{A}$  は  $\text{Game}_1$  の安全性ゲームで無視できない利得でゲームに勝利できると仮定する．PPTA シミュレータ  $\mathcal{S}$  は， $\mathcal{A}$  に対して  $\text{Game}_1$  を完璧にシミュレートし， $\mathcal{A}$  の Guess Phase での最終的な出力を利用して， $\Pi_{\text{KE}}$  に関する AA-IND-CCA 安全性ゲームに勝利しようとする． $\Pi_{\text{KE}}$  に関する AA-IND-CCA 安全性ゲームにおける挑戦者を  $\mathcal{CH}$  と表記する． $\mathcal{A}$ ， $\mathcal{S}$ ， $\mathcal{CH}$  の動作は以下の通りである．

**Setup Phase:**  $\mathcal{S}$  は， $\mathcal{CH}$  よりシステム公開鍵  $\text{PK}$  を受け取り， $\text{PK}$  を  $\mathcal{A}$  へ送る．

**Query Phase 1:**  $\mathcal{A}$  から  $\mathcal{S}$  への各オラクルクエリに対する  $\mathcal{S}$  の動作は以下の通りである．

**秘密鍵生成:**  $\mathcal{A}$  はアクセス構造  $\mathbb{A}$  をクエリする． $\mathcal{S}$  は  $\Pi_{\text{KE}}$  に関する AA-IND-CCA 安全性ゲームの Query Phase 1 の秘密鍵生成オラクルへのクエリとして  $\mathbb{A}$  を発行し， $\text{SK}$  を受け取る． $\mathcal{S}$  は  $\text{SK}$  を  $\mathcal{A}$  へ送る．

**鍵復号:**  $\mathcal{A}$  は鍵暗号文  $C_K$ ，アクセス構造  $\mathbb{A}$  をクエリする． $\mathcal{S}$  は Query Phase 1 の復号オラクルクエリとして  $(C_K, \mathbb{A})$  を発行し， $m / \perp =: \alpha$  を受け取る． $\mathcal{S}$  は  $\alpha$  を  $\mathcal{A}$  へ送る．

**Challenge Phase:**  $\mathcal{A}$  はターゲット復号者属性集合  $S_d^*$  を出力する． $\mathcal{S}$  は  $K_0, K_1 \xleftarrow{\mathcal{U}} \mathcal{K}$  を実行し，ターゲット平文，ターゲット復号者属性集合として， $(K_0, K_1, S_d^*)$  を  $\mathcal{CH}$  へ送り，チャレンジ暗号文  $C_K^*$  を受け取る． $\mathcal{S}$  は， $(K_1, C_K^*)$  を  $\mathcal{A}$  へ送る．

**Query Phase 2:**  $\mathcal{A}$  から  $\mathcal{S}$  への各オラクルクエリに対する  $\mathcal{S}$  の動作は以下の通りである．

**秘密鍵生成:** Query Phase 1 と同じ．

**鍵復号:** Query Phase 1 と同じ．

**Guess Phase:**  $\mathcal{A}$  チャレンジビット  $b$  に対する推測としてビット  $b'$  を出力し， $\mathcal{S}$  は  $b'$  を  $\mathcal{CH}$  に対して出力する．

$\Pi_{\text{KE}}$  に関する AA-IND-CCA 安全性ゲームのチャレンジビットを  $b$  とすると， $\mathcal{S}$  が  $\mathcal{A}$  に対して，チャレンジビットが  $b$  である  $\text{Game}_1$  を完璧にシミュレートできていることはほぼ自明であるため詳細な説明は割愛する．

$\mathcal{S}$  が  $\Pi_{\text{CK}}$  の AP-IND-CCA 安全性ゲームにおいてルール上禁止されたオラクルクエリを一度も行わないことに関しては，補題 4.1.2 と同様の説明で説明可能であるので，詳細な記述は割愛する．

以上より，以下の等式が成立する．

$$\text{Adv}_{\Pi_{\text{KE}}, S}^{\text{AA-IND-CCA}} = \text{Adv}_{\Pi_{\text{KK}}, \mathcal{A}}^{\text{Game}_1} = |\Pr[W_1] - \frac{1}{2}| \quad (6.5)$$

$|\Pr[W_1] - \frac{1}{2}|$ が無視できなくなるような PPTA  $\mathcal{A}$ が存在すると仮定すると，式 (6.5) より， $\text{Adv}_{\Pi_{\text{KE}}, S}^{\text{AA-IND-CCA}}$ は無視できなくなり，これは  $\Pi_{\text{KE}}$  が AA-IND-CCA 安全であるという事実と矛盾するので，その仮定は誤りである．従って， $\Pi_{\text{KE}}$  が AA-IND-CCA 安全ならば，いかなる PPTA 攻撃者  $\mathcal{A}$  に対しても， $|\Pr[W_1] - \frac{1}{2}|$ は無視できるほど小さい値になる．ゆえに，補題 6.1.2 が成立する．  $\square$

**定理 6.2 の証明** 証明の手順は定理 4.2 の証明と同様である．

KP-ABE 方式  $\Pi_{\text{KE}}$  に関して，復号者属性集合開示性の定義を満たすアルゴリズムを， $\text{Disclose}_{\text{KE}}$  と表記する．

全ての  $k$ ，全ての  $\mathcal{U}$ ，全ての  $(\text{PK}, \text{MK}) \leftarrow \text{KE.Setup}(1^k, \mathcal{U})$ ，全ての  $m$ ，全ての  $\mathbb{A}$ ，全ての  $S_d$ ，全ての  $K \xleftarrow{\mathcal{U}} \mathcal{K}$ ，全ての  $C_K \leftarrow \text{KE.Enc}(\text{PK}, K, S_d)$  に対して，アルゴリズム  $\text{Disclose}_{\text{KE}}$  をサブルーチンとして利用し， $C_K$  を入力変数とするアルゴリズム  $\text{Disclose}_{\text{KK}}$  を図 6.2 のように定義する．

$\text{Disclose}_{\text{KK}}(\text{PK}, C_K)$  :  
Return  $\text{Disclose}_{\text{KE}}(\text{PK}, C_K)$ .

図 6.2: 図 6.1 の KP-ABKEM 方式  $\Pi_{\text{KK}}$  の復号者属性集合開示性アルゴリズム  $\text{Disclose}_{\text{KK}}$

$\text{Disclose}_{\text{KK}}$  の定義より，次の等式が成立する．

$$\text{Disclose}_{\text{KK}}(\text{PK}, C_K) = \text{Disclose}_{\text{KE}}(\text{PK}, C_K) \quad (6.6)$$

$\text{Disclose}_{\text{KE}}$  は  $\Pi_{\text{KE}}$  の復号者属性集合開示性アルゴリズムだから，KP-ABE 方式の復号者属性集合開示性の定義より，次の等式が成立する．

$$\Pr[\text{Disclose}_{\text{KE}}(\text{PK}, C_K) = S_d] = 1 \quad (6.7)$$

式 (6.6), (6.7) より，次の等式が成立する．

$$\Pr[\text{Disclose}_{\text{KK}}(\text{PK}, C_K) = S_d] = 1 \quad (6.8)$$

式 (6.8) と KP-ABKEM 方式の復号者属性集合開示性の定義より，アルゴリズム  $\text{Disclose}_{\text{KK}}$  は復号者属性集合開示性アルゴリズムとしての条件を満たしている．よって，図 6.1 の KP-ABKEM 方式  $\Pi_{\text{KK}}$  は復号者属性集合開示性を満たす．ゆえに，定理 6.2 は成立する．  $\square$

# Chapter 7 鍵ポリシー型属性ベース Signcryptionの一般的構成

## 7.1 本章の概要

本章に示す成果は、鍵ポリシー型属性ベース暗号 (KP-ABKEM), 鍵ポリシー型属性ベース署名 (KP-ABS), データカプセル化メカニズム (DEM) を構成要素とする、鍵ポリシー型属性ベース Signcryption (KP-ABSC) の一般的構成法とその安全性証明である。具体的には、AA-IND-CCA 安全かつ復号者属性集合開示的な KP-ABKEM, AA-sEUF-CMA 安全かつ完全匿名かつ署名者属性集合衝突困難な KP-ABS, IND-CCA 安全かつ一対一対応な DEM から、AA-IND-CCA 安全かつ AA-sEUF-CMA 安全かつ完全匿名な KP-ABSC を一般的に構成できることを証明した。

なお、本成果 (鍵ポリシー型属性ベース Signcryption の一般的構成) の意義については、8.1.3 項で説明する。

また、本章の構成については以下の通りである。7.2 節で一般的構成法の説明を行う。7.3 節で安全性証明の詳述を行う。

## 7.2 提案する一般的構成法

構成要素として、KP-ABKEM 方式  $\Pi_{\text{KK}}: (\text{KK.Setup}, \text{KK.KeyGen}, \text{KK.Encap}, \text{KK.Decap})$ , DEM 方式  $\Pi_{\text{D}}: (\text{D.Encap}, \text{D.Decap})$ , KP-ABS 方式  $\Pi_{\text{KS}}: (\text{KS.Setup}, \text{KS.KeyGen}, \text{KS.Sig}, \text{KS.Ver})$  の三つの要素技術を用いた KP-ABSC の一般的構成法  $\Pi_{\text{KSC}}: (\text{KSC.Setup}, \text{KSC.KeyGen}_S, \text{KSC.KeyGen}_R, \text{KSC.SC}, \text{KSC.USC})$  を図 7.1 に示す。

## 7.3 安全性証明

図 7.1 の KP-ABSC 方式の一般的構成法  $\Pi_{\text{KSC}}$  の安全性は定理 7.1 及び定理 7.2, 定理 7.3 によって保障される。

**定理 7.1.** KP-ABKEM 方式  $\Pi_{\text{KK}}$  が AA-IND-CCA 安全であり、かつ DEM 方式  $\Pi_{\text{D}}$  が IND-CCA 安全であり、かつ KP-ABS 方式  $\Pi_{\text{KS}}$  が署名者属性集合衝突困難性を備えているならば、図 7.1 の KP-ABSC 方式  $\Pi_{\text{KSC}}$  は AA-IND-CCA 安全である。

**定理 7.2.** KP-ABS 方式  $\Pi_{\text{KS}}$  が AA-sEUF-CMA 安全であり、かつ DEM 方式  $\Pi_{\text{D}}$  が 1 対 1 対応性を有しており、かつ KP-ABKEM 方式が復号者属性集合開示性を有しているならば、図 7.1 の KP-ABSC 方式  $\Pi_{\text{KSC}}$  は AA-sEUF-CMA 安全である。

$\text{KSC.Setup}(1^k, \mathcal{U}_s, \mathcal{U}_r) :$ $(\text{PK}_{ss}, \text{MK}_{ss}) \leftarrow \text{KS.Setup}(1^k, \mathcal{U}_s);$ $(\text{PK}_{ck}, \text{MK}_{ck}) \leftarrow \text{KK.Setup}(1^k, \mathcal{U}_r);$ $\text{Return } (\text{PK}, \text{MK}) := ((\text{PK}_{ss}, \text{PK}_{ck}), (\text{MK}_{ss}, \text{MK}_{ck})).$
$\text{KSC.KeyGen}_s(\text{PK}, \text{MK}, \mathbb{A}_s) :$ $\text{Return } \text{SK}_s \leftarrow \text{KS.KeyGen}(\text{PK}_{ss}, \text{MK}_{ss}, \mathbb{A}_s).$
$\text{KSC.KeyGen}_r(\text{PK}, \text{MK}, \mathbb{A}_r) :$ $\text{Return } \text{SK}_r \leftarrow \text{KK.KeyGen}(\text{PK}_{ck}, \text{MK}_{ck}, \mathbb{A}_r).$
$\text{KSC.SC}(\text{PK}, m, \text{SK}_s, S_s, S_d) :$ $(K, C_K) \leftarrow \text{KK.Encap}(\text{PK}_{ck}, S_d);$ $\sigma \leftarrow \text{KS.Sig}(\text{PK}_{ss}, m \  C_K, \text{SK}_s, S_s);$ $C_D \leftarrow \text{D.Encap}(K, m \  \sigma);$ $\text{Return } C := (C_K, C_D).$
$\text{KSC.USC}(\text{PK}, C, \text{SK}_r, S_s) :$ $\text{Parse } C \text{ as } (C_K, C_D).$ $\alpha := \text{CK.Decap}(\text{PK}_{ck}, C_K, \text{SK}_r);$ $\text{If } \alpha = \perp, \text{ then return } \perp. \text{ Else } K := \alpha.$ $\beta := \text{D.Decap}(K, C_D);$ $\text{If } \beta = \perp, \text{ then return } \perp. \text{ Else } m \  \sigma := \beta.$ $\gamma := \text{SS.Ver}(\text{PK}_{ss}, \sigma, m \  C_K, S_s);$ $\text{If } \gamma = 0, \text{ then return } \perp. \text{ Else return } m.$

図 7.1: KP-ABSC 方式の一般的構成法  $\Pi_{\text{KSC}}$

**定理 7.3.**  $KP$ -ABS 方式  $\Pi_{KS}$  が完全匿名性を満たすならば、図 7.1 の  $KP$ -ABSC 方式  $\Pi_{KSC}$  は完全匿名性を満たす。

**定理 7.1 の証明** 証明の手順は定理 5.1 の証明と同様である。冗長的にならないよう、証明の詳細な記述を所々で省略する。

$\text{Game}_0, \text{Game}_1, \text{Game}_2$  の安全性ゲームを以下のように定義する。

**Game<sub>0</sub>:** PPTA 攻撃者  $\mathcal{A}$  と挑戦者  $\mathcal{CH}$  との間で行われる、 $\Pi_{KSC}$  に関する AA-IND-CCA 安全性ゲーム。

**Game<sub>1</sub>:**  $\text{Game}_0$  に以下の変更を加えたゲームとする。

- Query Phase 2 のアンサインクリプションオラクルにおいて、 $\mathcal{A}$  が発行するクエリ  $(C = (C_K, C_D), S_s, \mathbb{A}_r)$  に関して、 $C_K = C_K^*$  かつ  $C_D = C_D^*$  かつ  $S_s \neq S_s^*$  かつ  $S_d^* \in \mathbb{A}_r$  であれば、 $\mathcal{CH}$  は  $\perp$  を返答する。

**Game<sub>2</sub>:**  $\text{Game}_1$  に以下の変更を加えたゲームとする。

- Setup Phase において、 $\mathcal{CH}$  は  $K' \stackrel{U}{\leftarrow} \mathcal{K}$  を実行する。
- Challenge Phase において、 $\mathcal{CH}$  はチャレンジサインクリプテキスト  $C^* = (C_K^*, C_D^*)$  の第二成分  $C_D^*$  を鍵  $K'$  を用いて生成する。
- Query Phase 2 のアンサインクリプションオラクルにおいて、 $\mathcal{A}$  が発行するクエリ  $(C = (C_K, C_D), S_s, \mathbb{A}_r)$  に関して、 $C_K = C_K^*$  かつ  $C_D \neq C_D^*$  かつ  $S_d^* \in \mathbb{A}_r$  であれば、 $\mathcal{CH}$  は、 $C_D$  を鍵  $K'$  で復号する。

$\text{Game}_i (i = \{0, 1, 2\})$  の Guess Phase で攻撃者  $\mathcal{A}$  が正しい推測ビット  $b' = b$  を出力する事象を  $W_i$  と表記する。 $\text{Game}_0$  における攻撃者  $\mathcal{A}$  の優位性は定義より、

$$\begin{aligned} \text{Adv}_{\Pi_{KSC}, \mathcal{A}}^{\text{AA-IND-CCA}} &= \left| \Pr[W_0] - \frac{1}{2} \right| \\ &\leq |\Pr[W_0] - \Pr[W_1]| + |\Pr[W_1] - \Pr[W_2]| + \left| \Pr[W_2] - \frac{1}{2} \right| \quad (7.1) \end{aligned}$$

不等式 (7.1) と以下で証明する補題 7.1.1, 補題 7.1.2, 補題 7.1.3 より、 $\Pi_{KK}$  が AA-IND-CCA 安全であり、かつ  $\Pi_D$  が IND-CCA 安全であり、かつ  $\Pi_{KS}$  が署名者属性集合衝突困難性を備えているならば、いかなる PPTA 攻撃者  $\mathcal{A}$  に対しても、 $\text{Adv}_{\Pi_{KSC}, \mathcal{A}}^{\text{AA-IND-CCA}}$  はセキュリティパラメータ  $k$  に関して無視できるほど小さい値になる。ゆえに、定理 7.1 が成立する。□

**補題 7.1.1.**  $KP$ -ABS 方式  $\Pi_{KS}$  が署名者属性集合衝突困難性を備えるならば、いかなる PPTA 攻撃者  $\mathcal{A}$  に対しても、 $|\Pr[W_0] - \Pr[W_1]|$  はセキュリティパラメータ  $k$  に関して無視できるほど小さい値になる。

**補題 7.1.2.**  $KP$ - $ABKEM$  方式  $\Pi_{KK}$  が  $AA$ - $IND$ - $CCA$  安全ならば, いかなる  $PPTA$  攻撃者  $\mathcal{A}$  に対しても,  $|\Pr[W_1] - \Pr[W_2]|$  はセキュリティパラメータ  $k$  に関して無視できるほど小さい値になる.

**補題 7.1.3.**  $DEM$  方式  $\Pi_D$  が  $IND$ - $CCA$  安全ならば, いかなる  $PPTA$  攻撃者  $\mathcal{A}$  に対しても,  $|\Pr[W_2] - \frac{1}{2}|$  は無視できるほど小さい値になる.

以降では, 補題 7.1.1, 補題 7.1.2, 補題 7.1.3 の証明を行う.

**補題 7.1.1 の証明** 証明の手順は補題 5.1.1 の証明と同様である. 冗長的にならないよう, 証明の詳細な記述を所々で省略する.

以下に  $\text{Game}_0$  における攻撃者  $\mathcal{A}$  と挑戦者  $\mathcal{CH}$  の具体的なやり取りを示す.

**Setup Phase:**  $\mathcal{CH}$  は  $\text{KS.Setup}(1^k, \mathcal{U}_s) \rightarrow (\text{PK}_{ks}, \text{MK}_{ks})$ ,  $\text{KK.Setup}(1^k, \mathcal{U}_r) \rightarrow (\text{PK}_{kk}, \text{MK}_{kk})$  を実行し,  $\text{PK} := (\text{PK}_{ks}, \text{PK}_{kk})$  を  $\mathcal{A}$  に送る.

**Query Phase 1:**  $\mathcal{A}$  から  $\mathcal{CH}$  への各オラクルクエリに対する  $\mathcal{CH}$  の動作は以下の通りである.

**送信者用秘密鍵生成:**  $\mathcal{A}$  がアクセス構造  $\mathbb{A}_s$  をクエリする.  $\mathcal{CH}$  は  $\text{KS.KeyGen}(\text{PK}_{ss}, \text{MK}_{ss}, \mathbb{A}_s) \rightarrow \text{SK}_s$  を実行し  $\text{SK}_s$  を  $\mathcal{A}$  へ返す.

**受信者用秘密鍵生成:**  $\mathcal{A}$  がアクセス構造  $\mathbb{A}_r$  をクエリする.  $\mathcal{CH}$  は  $\text{KK.KeyGen}(\text{PK}_{kk}, \text{MK}_{kk}, \mathbb{A}_r) \rightarrow \text{SK}_r$  を実行し  $\text{SK}_r$  を  $\mathcal{A}$  に送る.

**サインクリプション:**  $\mathcal{A}$  が平文  $m$ , 署名者属性集合  $S_s$ , 復号者属性集合  $S_d$ , アクセス構造  $\mathbb{A}_s$  をクエリする.  $\mathcal{CH}$  は以下の処理を実行する.  $\text{KS.KeyGen}(\text{PK}_{ks}, \text{MK}_{ks}, \mathbb{A}_s) \rightarrow \text{SK}$ ,  $\text{KK.Encap}(\text{PK}_{kk}, S_d) \rightarrow (K, C_K)$ ,  $\text{KS.Sig}(\text{PK}_{ks}, m \| C_K, \text{SK}_s, S_s) \rightarrow \sigma$ ,  $\text{D.Encap}(K, m \| \sigma) \rightarrow C_D$ . そして,  $C := (C_K, C_D)$  を  $\mathcal{A}$  へ返す.

**アンサインクリプション:**  $\mathcal{A}$  がサインクリプテキスト  $C = (C_K, C_D)$ , 署名者属性集合  $S_s$ , アクセス構造  $\mathbb{A}_r$  をクエリする.  $\mathcal{CH}$  は  $\text{KK.KeyGen}(\text{PK}_{kk}, \text{MK}_{kk}, \mathbb{A}_r) \rightarrow \text{SK}_r$ ,  $\text{KK.Decap}(\text{PK}_{kk}, C_K, \text{SK}_r) =: x$  を実行する.  $x \neq \perp$  なら,  $\mathcal{S}$  は  $K := x$ ,  $\text{D.Decap}(K, C_D) =: y$  を実行する.  $y \neq \perp$  なら,  $\mathcal{CH}$  は  $m \| \sigma := y$ ,  $\text{KS.Ver}(\text{PK}_{ks}, \sigma, m \| C_K, S_s) =: z$  を実行し,  $z = 1$  なら,  $m$  を  $\mathcal{A}$  へ返す.  $x = \perp$  または  $y = \perp$  または  $z = 0$  なら,  $\perp$  を返す.

**Challenge Phase:**  $\mathcal{A}$  が長さの等しい平文  $m_0, m_1$ , ターゲット署名者属性集合  $S_s^*$ , ターゲット復号者属性集合  $S_d^*$ , ターゲット送信者アクセス構造  $\mathbb{A}_s^*$  を送る.  $\mathcal{CH}$  は以下の処理を実行する.  $\text{KS.KeyGen}(\text{PK}_{ks}, \text{MK}_{ks}, \mathbb{A}_s^*) \rightarrow \text{SK}$ ,  $b \xleftarrow{\text{U}} \{0, 1\}$ ,  $\text{KK.Encap}(\text{PK}_{kk}, S_d^*) \rightarrow (K^*, C_K^*)$ ,  $\text{KS.Sig}(\text{PK}_{ks}, m_b \| C_K^*, \text{SK}_s, S_s^*)$

$\rightarrow \sigma^*$ ,  $D.\text{Decap}(K^*, m_b \| \sigma^*) \rightarrow C_D^*$ . 結果的に  $\mathcal{CH}$  は  $C^* := (C_K^*, C_D^*)$  を  $\mathcal{A}$  へ送る.

**Query Phase 2:**  $\mathcal{A}$  から  $\mathcal{CH}$  への各オラクルクエリに対する  $\mathcal{CH}$  の動作は以下の通りである.

送信者用秘密鍵生成: Query Phase 1 と同じ.

受信者用秘密鍵生成: Query Phase 1 と同じ.

サインクリプション: Query Phase 1 と同じ.

アンサインクリプション: Query Phase 1 と同じ.

**Guess Phase:**  $\mathcal{A}$  はチャレンジビット  $b$  に対する推測として  $b'$  を  $\mathcal{CH}$  へ送る.

補題 5.1.1 及び補題 5.1.1.1 で用いた論理展開と同様の論理展開によって, 事象  $P'$  を以下に示す通りに定義すると, 事象  $P'$  の生起確率が無視できるほど小さくなるならば,  $|\Pr[W_0] - \Pr[W_1]|$  は無視できるほど小さくなる.

事象  $P'$

$\text{Game}_0$  において, Query Phase 2 のアンサインクリプションオラクルで,  $\mathcal{A}$  が  $C_K = C_K^*$  かつ  $C_D = C_D^*$  かつ  $S_s \neq S_s^*$  かつ  $S_d^* \in \mathbb{A}_r$  を満たし, かつ  $\text{KS.Ver}(\text{PK}_{k_s}, \sigma^*, m_b \| C_K^*, S_s) = 1$  を満たす,  $(C = (C_K, C_D), S_s, \mathbb{A}_r)$  を少なくとも一回クエリする.

ここで, 補題 7.1.1.1 より,  $\Pi_{\text{KS}}$  が署名者属性集合衝突困難性を備えるならば,  $\Pr[P']$  は無視できるほど小さい値になる. ゆえに, 補題 7.1.1 が成立する.  $\square$

**補題 7.1.1.1.**  $KP\text{-ABS}$  方式  $\Pi_{\text{KS}}$  が署名者属性集合衝突困難性を備えるならば, いかなる  $PPTA \mathcal{A}$  に対しても,  $\Pr[P']$  は無視できるほど小さい値になる.

以降では, 補題 7.1.1.1 の証明を示す.

**補題 7.1.1.1 の証明** 証明の手順は補題 5.1.1.1 と同様である.

$PPTA \mathcal{A}$  は  $\text{Game}_0$  の攻撃者であるとする. また, シミュレータ  $\mathcal{S}$  は  $\mathcal{A}$  に対し,  $\text{Game}_0$  を完璧にシミュレートし, 自身は  $\Pi_{\text{KS}}$  に関する署名者属性集合衝突困難性ゲームの攻撃者として動作する. また,  $\mathcal{CH}$  は  $\Pi_{\text{KS}}$  に関する署名者属性集合衝突困難性ゲームの挑戦者を表す. なお,  $\Pi_{\text{KS}}$  は署名者属性集合衝突困難性を備えるとする. さて,  $\mathcal{S}$  は以下の通りに動作するものとする.

**Setup Phase:**  $\mathcal{S}$  は  $\mathcal{CH}$  より  $\text{PK}_{k_s}$  を受け取る. そして,  $\text{KK.Setup}(1^k, \mathcal{U}_r) \rightarrow (\text{PK}_{kk}, \text{MK}_{kk})$  を実行し,  $\text{PK} := (\text{PK}_{k_s}, \text{PK}_{kk})$  を  $\mathcal{A}$  に送る.

**Query Phase 1:**  $\mathcal{A}$  が発行する各オラクルクエリに対する  $\mathcal{S}$  の動作は以下の通りである。

**送信者用秘密鍵生成:**  $\mathcal{A}$  がアクセス構造  $A_s$  をクエリする。  $\mathcal{S}$  は  $\mathcal{CH}$  へ秘密鍵生成オラクルクエリとして  $A_s$  を  $\mathcal{CH}$  へ送り,  $SK_s$  を受け取り,  $SK_s$  を  $\mathcal{A}$  へ送る。

**受信者用秘密鍵生成:**  $\mathcal{A}$  がアクセス構造  $A_r$  をクエリする。  $\mathcal{S}$  は  $KK.KeyGen(PK_{kk}, MK_{kk}, A_r) \rightarrow SK_r$  を実行し,  $SK_r$  を  $\mathcal{A}$  に送る。

**サインクリプション:**  $\mathcal{A}$  が平文  $m$ , 署名者属性集合  $S_s$ , 復号者属性集合  $S_d$ , アクセス構造  $A_s$  をクエリする。  $\mathcal{S}$  は  $\mathcal{CH}$  へ秘密鍵生成オラクルクエリとして  $A_s$  を送り,  $SK_s$  を受け取り, 以下の処理を実行する。  
 $KK.Encap(PK_{kk}, S_d) \rightarrow (K, C_K)$ ,  $KS.Sig(PK_{ks}, m || C_K, SK_s, S_s) \rightarrow \sigma$ ,  
 $D.Encap(K, m || \sigma) \rightarrow C_D$ . そして,  $C := (C_K, C_D)$  を  $\mathcal{A}$  へ返す。

**アンサインクリプション:**  $\mathcal{A}$  がサインクリプテキスト  $C = (C_K, C_D)$ , 署名者属性集合  $S_s$ , アクセス構造  $A_r$  をクエリする。  $\mathcal{S}$  は  $KK.KeyGen(PK_{kk}, MK_{kk}, A_r) \rightarrow SK_r$ ,  $KK.Decap(PK_{kk}, C_K, SK_r) =: x$  を実行する。  $x \neq \perp$  ならば,  $\mathcal{S}$  は  $K := x$ ,  $D.Decap(K, C_D) =: y$  を実行する。  $y \neq \perp$  ならば,  $\mathcal{S}$  は  $m || \sigma =: y$ ,  $KS.Ver(PK_{ks}, \sigma, m || C_K, S_s) =: z$  を実行する。  $\mathcal{S}$  は  $z = 1$  ならば,  $m$  を  $\mathcal{A}$  へ送り,  $x = \perp$  または  $y = \perp$  または  $z = 0$  ならば,  $\perp$  を  $\mathcal{A}$  へ送る。

**Challenge Phase:**  $\mathcal{A}$  が長さの等しい平文  $m_0, m_1$ , ターゲット署名者属性集合  $S_s^*$ , ターゲット復号者属性集合  $S_d^*$ , ターゲット送信者アクセス構造  $A_s^*$  を送る。  $\mathcal{S}$  は  $b \xleftarrow{U} \{0, 1\}$ ,  $KK.Encap(PK_{kk}, S_d^*) \rightarrow (K^*, C_K^*)$  を実行する。 続けて,  $\mathcal{S}$  は  $\mathcal{CH}$  へ  $(m_b || C_K^*, A_s^*, S_s^*)$  を送り,  $\sigma^*$  を受け取る。 続けて,  $\mathcal{S}$  は,  $D.Encap(K^*, m_b || \sigma^*) \rightarrow C_D^*$  を実行し,  $C^* := (C_K^*, C_D^*)$  を  $\mathcal{A}$  へ返す。

**Query Phase 2:**  $\mathcal{A}$  から  $\mathcal{S}$  への各オラクルクエリに対する  $\mathcal{S}$  の動作は以下の通りである。

**送信者用秘密鍵生成:** Query Phase 1 と同じ。

**受信者用秘密鍵生成:** Query Phase 1 と同じ。

**サインクリプション:** Query Phase 1 と同じ。

**アンサインクリプション:**  $\mathcal{A}$  が  $(C_K, C_D)$ ,  $S_s$ ,  $A_r$  をクエリする。

- (I)  $C_K = C_K^*$ , かつ  $C_D = C_D^*$ , かつ  $S_s \neq S_s^*$ , かつ  $S_d^* \in A_r$  の場合,  $\mathcal{S}$  は  $KS.Ver(PK_{ks}, \sigma^*, m_b || C_K^*, S_s) =: z$  を実行する。  $z = 1$  ならば,  $\mathcal{S}$  は  $\mathcal{CH}$  に対して, 署名者属性集合衝突困難性ゲームの Output Phase で  $S_s$  を出力して,  $\mathcal{A}$  に対する  $\text{Game}_0$  のシミュレーションはこの時点で強制的に終了する。  $z = 0$  ならば,  $\mathcal{S}$  は  $\mathcal{A}$  に  $\perp$  を

返す.

(II) その他の場合,  $S$  は Query Phase 1 での動作と同じ動作を行う.

**Guess Phase:**  $\mathcal{A}$  はチャレンジビット  $b$  に対する推測として  $b'$  を  $CH$  へ送る.

$S$  は  $\mathcal{A}$  に対し,  $\text{Game}_0$  を完璧にシミュレートできていることは自明であり, 詳細な説明は割愛する. この状況で事象  $P'$  が生じた場合に  $S$  が署名者属性集合衝突困難性ゲームに勝利できることは自明である. 従って, 次の等式が成立する.

$$\Pr[P'] = \text{Adv}_{\Pi_{KS}, S}^{\text{SASCR}} \quad (7.2)$$

式 (7.2) より,  $\Pr[P']$  が無視できなくなると仮定すると,  $\text{Adv}_{\Pi_{KS}, S}^{\text{SASCR}}$  が無視できなくなる. だが, これは  $\Pi_{KS}$  が署名者属性集合衝突困難性を備えるという事実に矛盾しているため, 背理法より仮定が誤りである. 従って, 補題 7.1.1.1 は成立する.  $\square$

**補題 7.1.2 の証明** 証明の手順は補題 5.1.2 の証明と同様である. 冗長的にならないよう, 証明の詳細な記述を所々で省略する.

補題 5.1.2 の証明内においては  $CH$  は KP-ABKEM 方式  $\Pi_{KK}$  に関する AA-IND-CCA 安全性ゲームにおける挑戦者を意味するものとする. なお,  $\Pi_{KK}$  は AA-IND-CCA 安全であるとする. PPTA 攻撃者  $\mathcal{A}$  は  $\text{Game}_1$ ,  $\text{Game}_2$  それぞれのゲームにおける攻撃者として動作する. シミュレータ  $S$  は  $\Pi_{KK}$  に関する AA-IND-CCA ゲームにおける攻撃者として動作する.  $S$  は  $\Pi_{KK}$  に関する AA-IND-CCA ゲームにおいて  $CH$  側で決定されるチャレンジビット  $\beta$  は知らない状況で, チャレンジ鍵及びチャレンジ鍵暗号文 ( $K_\beta$ ,  $C_K^*$ ) を利用して,  $\mathcal{A}$  に対して,  $\beta = 1$  の場合に  $\text{Game}_1$  を,  $\beta = 0$  の場合に  $\text{Game}_2$  を完全に正しくシミュレートしたい. そのために,  $S$  は以下のように動作する.

**Setup Phase:**  $S$  は  $CH$  から  $PK_{kk}$  を受取る.  $S$  は  $KS.\text{Setup}(1^k, \mathcal{U}_s) \rightarrow (PK_{ks}, MK_{ks})$ ,  $PK := (PK_{ks}, PK_{kk})$  を実行し,  $PK$  を  $\mathcal{A}$  へ送る.

**Query Phase 1:**  $\mathcal{A}$  から  $S$  への各オラクルクエリに対する  $S$  の動作は以下の通りである.

**送信者用秘密鍵生成:**  $\mathcal{A}$  がアクセス構造  $A_s$  をクエリする.  $S$  は  $SS.\text{KeyGen}(PK_{ks}, MK_{ks}, A_s) \rightarrow SK_s$  を実行し  $SK_s$  を  $\mathcal{A}$  へ返す.

**受信者用秘密鍵生成:**  $\mathcal{A}$  がアクセス構造  $A_r$  をクエリする.  $S$  は  $CH$  へ秘密鍵生成オラクルクエリとして  $A_r$  を送り, 鍵  $SK_r$  を受け取る.  $S$  は  $\mathcal{A}$  へ  $SK_r$  を返す.

**サインクリプション:**  $\mathcal{A}$  が平文  $m$ , 署名者属性集合  $S_s$ , 復号者属性集合  $S_d$ , アクセス構造  $A_s$  をクエリする.  $SK_s := SS.\text{KeyGen}(PK_{ks}, MK_{ks},$

$\mathbb{A}_s$ )として,  $\mathcal{S}$ は以下の処理を実行する.  $\text{KK.Encap}(\text{PK}_{kk}, S_d) \rightarrow (K, C_K)$ ,  $\text{KS.Sig}(\text{PK}_{ks}, m \| C_K, \text{SK}_s, S_s) \rightarrow \sigma$ ,  $\text{D.Encap}(K, m \| \sigma) \rightarrow C_D$ . そして,  $C := (C_K, C_D)$ を  $\mathcal{A}$ へ返す.

**アンサインクリプション:**  $\mathcal{A}$ がサインクリプテキスト  $C = (C_K, C_D)$ , 署名者属性集合  $S_s$ , アクセス構造  $\mathbb{A}_r$ をクエリする.  $\mathcal{S}$ は  $(C_K, \mathbb{A}_r)$ を  $\mathcal{CH}$ へ鍵復号オラクルクエリとして送り,  $x$ を受け取る.  $x \neq \perp$ なら,  $\mathcal{S}$ は  $K := x$ ,  $\text{D.Decap}(K, C_D) := y$ を実行する.  $y \neq \perp$ なら,  $\mathcal{S}$ は  $m \| \sigma := y$ ,  $\text{KS.Ver}(\text{PK}_{ks}, \sigma, m \| C_K, S_s) \rightarrow z$ を実行し,  $z = 1$ なら,  $m$ を  $\mathcal{A}$ へ返す.  $x = \perp$ または  $y = \perp$ または  $z = 0$ なら,  $\perp$ を返す.

**Challenge Phase:**  $\mathcal{A}$ が長さの等しい平文  $m_0, m_1$ , ターゲット署名者属性集合  $S_s^*$ , ターゲット復号者属性集合  $S_d^*$ , ターゲット送信者アクセス構造  $\mathbb{A}_s^*$ を送る.  $\text{SK}_s := \text{KS.KeyGen}(\text{PK}_{ks}, \text{MK}_{ks}, \mathbb{A}_s^*)$ として,  $\mathcal{S}$ は最初に  $b \xleftarrow{\text{U}} \{0, 1\}$ を実行する. 続けて,  $S_d^*$ を  $\Pi_{\text{KK}}$ に関する AA-IND-CCA ゲームのターゲット復号者属性集合として  $\mathcal{CH}$ へ送り,  $(K_\beta, C_K^*)$ を受取る. 続けて以下の処理を実行する.  $\text{KS.Sig}(\text{PK}_{ks}, m_b \| C_K^*, \text{SK}_s, S_s^*) \rightarrow \sigma^*$ ,  $\text{D.Decap}(K^*, m_b \| \sigma^*) \rightarrow C_D^*$ . 結果的に  $\mathcal{S}$ は  $C^* := (C_K^*, C_D^*)$ を  $\mathcal{A}$ へ送る.

**Query Phase 2:**  $\mathcal{A}$ から  $\mathcal{S}$ への各オラクルクエリに対する  $\mathcal{S}$ の動作は以下の通りである.

**送信者用秘密鍵生成:** Query Phase 1 と同じ.

**受信者用秘密鍵生成:** Query Phase 1 と同じ.

**サインクリプション:** Query Phase 1 と同じ.

**アンサインクリプション:**  $\mathcal{A}$ が  $(C_K, C_D)$ ,  $S_s$ ,  $\mathbb{A}_r$ をクエリする.

- (I)  $C_K = C_K^*$ , かつ  $C_D \neq C_D^*$ , かつ  $S_d^* \in \mathbb{A}_r$  の場合,  $\mathcal{S}$ は,  $\text{D.Decap}(K_\beta, C_D) := y$ を実行し,  $y \neq \perp$ ならば,  $m \| \sigma := y$ ,  $\text{KS.Ver}(\text{PK}_{ks}, \sigma, m \| C_K, S_s) \rightarrow z$ を実行し,  $z = 1$ ならば  $m$ を返す.  $y = \perp$ または  $z = 0$ ならば,  $\perp$ を返す.
- (II)  $C_K = C_K^*$ , かつ  $C_D = C_D^*$ , かつ  $S_s \neq S_s^*$ , かつ  $S_d^* \in \mathbb{A}_r$  の場合,  $\mathcal{S}$ は  $\perp$ を返す.
- (III) その他の場合,  $\mathcal{S}$ は  $(C_K, \mathbb{A}_r)$ を  $\mathcal{CH}$ へ鍵復号オラクルクエリとして送り,  $x$ を受け取る.  $x \neq \perp$ であれば,  $K := x$ ,  $\text{D.Decap}(K, C_D) := y$ を実行し,  $y \neq \perp$ ならば  $m \| \sigma := y$ ,  $\text{KS.Ver}(\text{PK}_{ks}, \sigma, m \| C_K, S_s) := z$ を実行し,  $z = 1$ ならば  $m$ を返す.  $x = \perp$ または  $y = \perp$ または  $z = 0$ ならば,  $\perp$ を返す.

**Guess Phase:**  $\mathcal{A}$ はチャレンジビット  $b$ に対する推測ビットとして  $b'$ を  $\mathcal{S}$ へ送る.  $\mathcal{S}$ は  $b' = b$ ならば  $\beta' := 1$ ,  $b' \neq b$ ならば  $\beta' := 0$ を実行し,  $\beta'$ を  $\Pi_{\text{KK}}$ に関する AA-IND-CCA ゲームのチャレンジビット  $\beta$ に対する推測ビット

として  $\mathcal{CH}$  へ送る.

補題 5.1.2 の証明と同様に, 次の等式を導出できる.

$$\text{Adv}_{\Pi_{\text{KK}}, \mathcal{S}}^{\text{AA-IND-CCA}} = \frac{1}{2} |\Pr[b' = b | \beta = 1] - \Pr[b' = b | \beta = 0]| \quad (7.3)$$

$\beta = 1$  (resp.  $\beta = 0$ ) の場合に, シミュレータ  $\mathcal{S}$  が攻撃者  $\mathcal{A}$  に対して,  $\text{Game}_1$  (resp.  $\text{Game}_2$ ) の挑戦者としての応答を完璧にシミュレートできている事, そして  $\mathcal{S}$  が  $\mathcal{CH}$  に対して  $\Pi_{\text{KK}}$  に関する AA-IND-CCA ゲームのルール上禁止されているオラクルクエリを一度も行っていない事に関しては, 補題 5.1.2 の証明で用いた説明と同様の説明で説明可能であるので, 詳細な記述は割愛する.

従って, 次の 2 つの等式が成り立つ.

$$\Pr[b' = b | \beta = 1] = \Pr[W_1] \quad (7.4)$$

$$\Pr[b' = b | \beta = 0] = \Pr[W_2] \quad (7.5)$$

等式 (7.3), (7.4), (7.5) より, 以下の等式が成り立つ.

$$\text{Adv}_{\Pi_{\text{KK}}, \mathcal{S}}^{\text{AA-IND-CCA}} = \frac{1}{2} |\Pr[W_1] - \Pr[W_2]| \quad (7.6)$$

$|\Pr[W_1] - \Pr[W_2]|$  が無視できなくなるような  $\mathcal{A}$  が存在すると仮定すると, 式 (7.6) より,  $\text{Adv}_{\Pi_{\text{KK}}, \mathcal{S}}^{\text{AA-IND-CCA}}$  は無視できなくなり, これは  $\Pi_{\text{KK}}$  が AA-IND-CCA 安全であるという事実に矛盾するので, その仮定は誤りである. ゆえに,  $\Pi_{\text{KK}}$  が AA-IND-CCA 安全ならば, いかなる PPTA 攻撃者  $\mathcal{A}$  に対しても,  $|\Pr[W_1] - \Pr[W_2]|$  はセキュリティパラメータ  $k$  に関して無視できるほど小さい値になる.  $\square$

**補題 7.1.3 の証明** 証明の手順は補題 5.1.3 の証明と同様である. 冗長的にならないよう, 証明の詳細な記述を所々で省略する.

補題 7.1.3 の証明内においては,  $\mathcal{CH}$  は, DEM 方式  $\Pi_{\text{D}}$  に関する IND-CCA 安全性ゲームにおける挑戦者を意味する. 但し,  $\Pi_{\text{D}}$  は IND-CCA 安全であるとする.  $\mathcal{S}$  は  $\mathcal{A}$  に対して,  $\Pi_{\text{KK}}$  に関する  $\text{Game}_2$  の安全性ゲームを完璧にシミュレートし  $\mathcal{A}$  の最終的な出力を利用して  $\Pi_{\text{D}}$  に関する IND-CCA 安全性を破ろうとする PPTA であり,  $\mathcal{S}$  の動作を以下のように定める.

**Setup Phase:**  $\mathcal{S}$  は  $\text{KS.Setup}(1^k, \mathcal{U}_s) \rightarrow (\text{PK}_{ks}, \text{MK}_{ks}), \text{KK.Setup}(1^k, \mathcal{U}_r) \rightarrow (\text{PK}_{kk}, \text{MK}_{kk})$  を計算し,  $\text{PK} := (\text{PK}_{ks}, \text{PK}_{kk})$  を  $\mathcal{A}$  に送る.

**Query Phase 1:**  $\mathcal{A}$  から  $\mathcal{S}$  への各オラクルクエリに対する  $\mathcal{S}$  の動作は以下の通りである.

**送信者用秘密鍵生成:**  $\mathcal{A}$  がアクセス構造  $\mathbb{A}_s$  をクエリする.  $\mathcal{S}$  は  $\text{KS.KeyGen}(\text{PK}_{ks}, \text{MK}_{ks}, \mathbb{A}_s) \rightarrow \text{SK}_s$  を実行し  $\text{SK}_s$  を  $\mathcal{A}$  へ返す.

**受信者用秘密鍵生成:**  $\mathcal{A}$  がアクセス構造  $\mathbb{A}_r$  をクエリする.  $\mathcal{S}$  は  $\text{KK.KeyGen}(\text{PK}_{kk}, \text{MK}_{kk}, \mathbb{A}_r) \rightarrow \text{SK}_r$  を実行し  $\text{SK}_r$  を  $\mathcal{A}$  に送る.

**サインクリプション:**  $\mathcal{A}$  が平文  $m$ , 署名者属性集合  $S_s$ , 復号者属性集合  $S_d$ , アクセス構造  $\mathbb{A}_s$  をクエリする.  $\text{SK}_s := \text{KS.KeyGen}(\text{PK}_{ks}, \text{MK}_{ks}, \mathbb{A}_s)$  として,  $\mathcal{S}$  は以下の処理を実行する.  $\text{KK.Encap}(\text{PK}_{kk}, S_d) \rightarrow (K, C_K)$ ,  $\text{KS.Sig}(\text{PK}_{ks}, m \| C_K, \text{SK}_s, S_s) \rightarrow \sigma$ ,  $\text{D.Encap}(K, m \| \sigma) \rightarrow C_D$ . そして,  $C := (C_K, C_D)$  を  $\mathcal{A}$  へ返す.

**アンサインクリプション:**  $\mathcal{A}$  がサインクリプテキスト  $C = (C_K, C_D)$ , 署名者属性集合  $S_s$ , アクセス構造  $\mathbb{A}_r$  をクエリする.  $\mathcal{S}$  は  $\text{KK.KeyGen}(\text{PK}_{kk}, \text{MK}_{kk}, \mathbb{A}_r) \rightarrow \text{SK}_r$ ,  $\text{KK.Decap}(\text{PK}_{kk}, C_K, \text{SK}_r) =: x$  を実行する.  $x \neq \perp$  なら,  $\mathcal{S}$  は  $K := x$ ,  $\text{D.Decap}(K, C_D) =: y$  を実行する.  $y \neq \perp$  なら,  $\mathcal{S}$  は  $m \| \sigma := y$ ,  $\text{KS.Ver}(\text{PK}_{ks}, \sigma, m \| C_K, S_s) =: z$  を実行し,  $z = 1$  なら,  $m$  を  $\mathcal{A}$  へ返す.  $x = \perp$  または  $y = \perp$  または  $z = 0$  なら,  $\perp$  を返す.

**Challenge Phase:**  $\mathcal{A}$  が長さの等しい平文  $m_0, m_1$ , ターゲット署名者属性集合  $S_s^*$ , ターゲット復号者属性集合  $S_d^*$ , ターゲット送信者アクセス構造  $\mathbb{A}_s^*$  を送る.  $\mathcal{S}$  は  $\text{SK}_s := \text{KS.KeyGen}(\text{PK}_{ks}, \text{MK}_{ks}, \mathbb{A}_s^*)$  として, 以下の処理を実行する.  $\text{KK.Encap}(\text{PK}_{kk}, S_d^*) \rightarrow (K^*, C_K^*)$ ,  $\text{KS.Sig}(\text{PK}_{ks}, m_0 \| C_K^*, \text{SK}_s, S_s^*) \rightarrow \sigma_0$ ,  $\text{KS.Sig}(\text{PK}_{ks}, m_1 \| C_K^*, \text{SK}_s, S_s^*) \rightarrow \sigma_1$ ,  $M_0 := m_0 \| \sigma_0$ ,  $M_1 := m_1 \| \sigma_1$ .  $\mathcal{S}$  は平文  $M_0, M_1$  を  $\mathcal{CH}$  へ送り,  $C_D^*$  を受け取り,  $C^* := (C_K^*, C_D^*)$  を  $\mathcal{A}$  へ送る.

**Query Phase 2:**  $\mathcal{A}$  から  $\mathcal{S}$  への各オラクルクエリに対する  $\mathcal{S}$  の動作は以下の通りである.

**送信者用秘密鍵生成:** Query Phase 1 と同じ.

**受信者用秘密鍵生成:** Query Phase 1 と同じ.

**サインクリプション:** Query Phase 1 と同じ.

**アンサインクリプション:**  $\mathcal{A}$  が  $C = (C_K, C_D), S_s, \mathbb{A}_r$  をクエリする.

- (I)  $C_K = C_K^*$ , かつ  $C_D \neq C_D^*$ , かつ  $S_d^* \in \mathbb{A}_r$  の場合,  $\mathcal{S}$  は  $C_D$  をデータ復号オラクルクエリとして  $\mathcal{CH}$  へ送り,  $y$  を受け取る.  $\mathcal{S}$  は,  $y \neq \perp$  ならば  $m \| \sigma := y$ ,  $\text{KS.Ver}(\text{PK}_{ks}, \sigma, m \| C_K^*, S_s) \rightarrow z$  を実行し,  $z = 1$  なら  $m$  を  $\mathcal{A}$  へ送る.  $y = \perp$  または  $z = 0$  なら,  $\perp$  を送る.
- (II)  $C_K = C_K^*$ , かつ  $C_D = C_D^*$ , かつ  $S_s \neq S_s^*$ , かつ  $S_d^* \in \mathbb{A}_r$  の場合,  $\mathcal{S}$  は  $\perp$  を返す.
- (III) その他の場合,  $\mathcal{S}$  は Query Phase 1 でアンサインクリプションオ

ラクルクエリが発行された時の動作と同様の動作を行う。

**Guess Phase:**  $\mathcal{A}$  はチャレンジビット  $b$  に関する推測として  $b'$  を  $\mathcal{S}$  へ送る.  $\mathcal{S}$  はチャレンジビット  $\beta$  に関する推測として  $\beta' := b'$  を  $\mathcal{CH}$  へ送る.

シミュレータ  $\mathcal{S}$  が攻撃者  $\mathcal{A}$  に対して,  $\text{Game}_2$  の挑戦者としての応答を完璧にシミュレート出来ている事, そして  $\mathcal{S}$  が  $\mathcal{CH}$  に対して  $\Pi_D$  に関する IND-CCA ゲームでルール上禁止されているオラクルクエリを一度も行っていない事に関しては, 補題 5.1.3 で用いた説明と同様の説明で説明可能であるので, 詳細な説明の記述は省略する.

従って,  $\mathcal{A}$  の  $\text{Game}_2$  における優位性を  $\text{Adv}_{\Pi_{\text{KSC}}, \mathcal{A}}^{\text{Game}_2}$  とすれば, 定義より,

$$\text{Adv}_{\Pi_{\text{KSC}}, \mathcal{A}}^{\text{Game}_2} = |\Pr[b' = b] - \frac{1}{2}| = |\Pr[W_2] - \frac{1}{2}| \quad (7.7)$$

さらに,  $b = \beta$ ,  $b' = \beta'$  なので,  $b = b'$  ならば  $\beta = \beta'$  となるし,  $\beta = \beta'$  ならば  $b = b'$  なる. よって,  $b' = b$  という事象が生起する確率と  $\beta' = \beta$  という事象が生起する確率は等しく, 以下の等式が成り立つ.

$$\Pr[b' = b] = \Pr[\beta' = \beta] \quad (7.8)$$

DEM 方式  $\Pi_D$  に関する IND-CCA ゲームにおける  $\mathcal{S}$  の優位性の式と, 等式 (7.7), 等式 (7.8) より, 以下の等式が成り立つ.

$$\text{Adv}_{\Pi_D, \mathcal{S}}^{\text{IND-CCA}} = |\Pr[\beta' = \beta] - \frac{1}{2}| = |\Pr[b' = b] - \frac{1}{2}| = |\Pr[W_2] - \frac{1}{2}| \quad (7.9)$$

$|\Pr[W_2] - \frac{1}{2}|$  が無視できなくなるような  $\mathcal{A}$  が存在すると仮定すると, 式 (7.9) より,  $\text{Adv}_{\Pi_D, \mathcal{S}}^{\text{IND-CCA}}$  は無視できなくなり, これは  $\Pi_D$  が IND-CCA 安全であるという事実に矛盾するので, その仮定は誤りである. ゆえに,  $\Pi_D$  が IND-CCA 安全ならば, いかなる PPTA 攻撃者  $\mathcal{A}$  に対しても,  $|\Pr[W_2] - \frac{1}{2}|$  はセキュリティパラメータ  $k$  に関して無視できるほど小さい値になる.  $\square$

**定理 7.2 の証明** 証明の手順は定理 5.2 の証明と同様である. 冗長的にならないよう, 証明の詳細な記述を所々で省略する.

補題 7.2.1, 補題 7.2.2 により, 定理 7.2 は成立する.  $\square$

**補題 7.2.1.**  $KP\text{-}ABKEM$  方式  $\Pi_{\text{KK}}$  が復号者属性集合開示性を満たすならば, 図 7.1 の  $KP\text{-}ABSC$  方式  $\Pi_{\text{KSC}}$  は復号者属性集合開示性を満たす.

**補題 7.2.2.**  $KP\text{-}ABS$  方式  $\Pi_{\text{KS}}$  が  $AA\text{-}s\text{EUF}\text{-}CMA$  安全であり,  $DEM$  方式  $\Pi_D$  が一対一対応性を満たし,  $KP\text{-}ABKEM$  方式  $\Pi_{\text{KK}}$  が復号者属性集合開示性を満たすのならば, いかなる PPTA  $\mathcal{A}$  に対しても,  $\text{Adv}_{\Pi_{\text{KSC}}, \mathcal{A}}^{\text{AA-sEUFCMA}}$  はセキュリティパラメータ  $k$  に関して, 無視できるほど小さい値になる.

以降, 補題 7.2.1, 補題 7.2.2 の証明を示す.

**補題 7.2.1 の証明** 証明の手順は補題 5.2.1 の証明と同様である.

全ての  $k$ , 全ての  $\mathcal{U}_s$ , 全ての  $\mathcal{U}_r$ , 全ての  $(\text{PK}_{ks}, \text{MK}_{ks}) \leftarrow \text{KS.Setup}(1^k, \mathcal{U}_s)$ , 全ての  $(\text{PK}_{kk}, \text{MK}_{kk}) \leftarrow \text{KK.Setup}(1^k, \mathcal{U}_r)$ , 全ての  $m$ , 全ての  $\mathbb{A}_s$ , 全ての  $\text{SK}_s \leftarrow \text{KS.KeyGen}(\text{PK}, \text{MK}, \mathbb{A}_s)$ , 全ての  $S_s (s.t. S_s \in \mathbb{A}_s)$ , 全ての  $S_d$ , 全ての  $(K, C_K) \leftarrow \text{KK.Encap}(\text{PK}_{kk}, S_d)$ , 全ての  $\sigma \leftarrow \text{KS.Sig}(\text{PK}_{ks}, m \| C_K, \text{SK}_s, S_s)$ , 全ての  $C_D \leftarrow \text{D.Encap}(K, m \| \sigma)$  に対して, ここで  $\text{PK} := (\text{PK}_{ks}, \text{PK}_{kk})$ , また  $C := (C_K, C_D)$  として, アルゴリズム  $\text{Disclose}_{\text{KK}}$  をサブルーチンとして利用し,  $C$  を入力変数とするアルゴリズム  $\text{Disclose}_{\text{KSC}}$  を図 7.2 のように定義する.

$\text{Disclose}_{\text{KSC}}(\text{PK}, C) :$   
 Parse  $\text{PK}$  as  $(\text{PK}_{ks}, \text{PK}_{kk})$ .  
 Parse  $C$  as  $(C_K, C_D)$ .  
 Return  $\text{Disclose}_{\text{KK}}(\text{PK}_{kk}, C_K)$ .

図 7.2: KP-ABSC 方式  $\Pi_{\text{KSC}}$  の復号者属性集合開示性アルゴリズム  $\text{Disclose}_{\text{KSC}}$

図 7.2 の定義より, 次の等式が成立する.

$$\text{Disclose}_{\text{KSC}}(\text{PK}, C) = \text{Disclose}_{\text{KK}}(\text{PK}_{kk}, C_K) \quad (7.10)$$

$\text{Disclose}_{\text{KK}}$  は  $\Pi_{\text{KK}}$  の復号者属性集合開示性アルゴリズムだから, KP-ABKEM 方式の復号者属性集合開示性の定義より, 次の等式が成立する.

$$\Pr[\text{Disclose}_{\text{KK}}(\text{PK}_{kk}, C_K) = S_d] = 1 \quad (7.11)$$

式 (7.10), (7.11) より, 次の等式が成立する.

$$\Pr[\text{Disclose}_{\text{KSC}}(\text{PK}, C) = S_d] = 1 \quad (7.12)$$

式 (7.12) と KP-ABSC 方式の復号者属性集合開示性の定義より, アルゴリズム  $\text{Disclose}_{\text{KSC}}$  は復号者属性集合開示性アルゴリズムとしての条件を満たしている. よって, 図 7.1 の KP-ABSC 方式  $\Pi_{\text{KSC}}$  は復号者属性集合開示性を満たす. ゆえに, 補題 7.2.1 は成立する.  $\square$

**補題 7.2.2 の証明** 補題 7.2.2 の証明内においては, PPTA  $\mathcal{A}$  は図 7.1 の KP-ABSC 方式  $\Pi_{\text{KSC}}$  に関する AA-sEUF-CMA 安全性ゲームにおける攻撃者を意味する. 但し,  $\mathcal{A}$  は当該安全性を無視できない優位性で破ることができると仮定する. また,  $\mathcal{CH}$  は KP-ABS 方式  $\Pi_{\text{KS}}$  に関する AA-sEUF-CMA 安全性ゲームにおける挑戦者を意味する. 但し,  $\Pi_{\text{KS}}$  は AA-sEUF-CMA 安全な KP-ABS 方式であるとする.  $\mathcal{S}$  は  $\mathcal{A}$  に対して  $\Pi_{\text{KSC}}$  に関する AA-sEUF-CMA 安全性ゲームを完璧にシミュレートし,  $\mathcal{A}$  の最終的な出力を利用して,  $\Pi_{\text{KS}}$  に関する AA-sEUF-CMA 安全性ゲームに勝利しようとするシミュレータを意味し,  $\mathcal{S}$  の動作を以下のように定める.

**Setup Phase:**  $\mathcal{S}$  は  $\mathcal{CH}$  より  $\text{PK}_{ks}$  を受取る.  $\mathcal{S}$  は  $\text{KK.Setup}(1^k, \mathcal{U}_r) \rightarrow (\text{PK}_{kk}, \text{MK}_{kk}), \text{PK} := (\text{PK}_{ks}, \text{PK}_{kk})$  を実行し,  $\text{PK}$  を  $\mathcal{A}$  へ送る.

**Query Phase:**  $\mathcal{A}$  から  $\mathcal{S}$  への各オラクルクエリに対する  $\mathcal{S}$  の動作は以下の通りである.

**送信者用秘密鍵生成:**  $\mathcal{A}$  がアクセス構造  $\mathbb{A}_s$  をクエリする.  $\mathcal{S}$  は  $\mathbb{A}_s$  を  $\mathcal{CH}$  へ鍵生成オラクルクエリとして送り,  $\text{SK}_s$  を受け取り,  $\text{SK}_s$  を  $\mathcal{A}$  へ送る.

**受信者用秘密鍵生成:**  $\mathcal{A}$  がアクセス構造  $\mathbb{A}_r$  をクエリする.  $\mathcal{S}$  は  $\text{KK.KeyGen}(\text{PK}_{kk}, \text{MK}_{kk}, \mathbb{A}_r) \rightarrow \text{SK}_r$  を実行し,  $\text{SK}_r$  を  $\mathcal{A}$  へ送る.

**サインクリプション:**  $\mathcal{A}$  が平文  $m$ , 署名者属性集合  $S_s$ , 復号者属性集合  $S_d$ , アクセス構造  $\mathbb{A}_s$  (s.t.  $S_s \in \mathbb{A}_s$ ) をクエリする. 安全性証明の便宜上, この時点でリスト  $\mathcal{L}_{\text{SC},ksc}$  の要素数を  $i-1$  として,  $\mathcal{A}$  から送られた先述のクエリをそれぞれ以下のように表記する.  $m_{\text{SC},ksc}^{(i)} := m$ ,  $S_s^{(i)}_{\text{SC},ksc} := S_s$ ,  $S_d^{(i)}_{\text{SC},ksc} := S_d$ ,  $\mathbb{A}_s^{(i)}_{\text{SC},ksc} := \mathbb{A}_s$ .  $\mathcal{S}$  は  $\text{KK.Encap}(\text{PK}_{kk}, S_d^{(i)}_{\text{SC},ksc}) =: (K_{\text{SC},ksc}^{(i)}, C_{K_{\text{SC},ksc}^{(i)}}^{(i)})$  を実行し,  $(m_{\text{SC},ksc}^{(i)} \| C_{K_{\text{SC},ksc}^{(i)}}^{(i)}, \mathbb{A}_s^{(i)}_{\text{SC},ksc}, S_s^{(i)}_{\text{SC},ksc})$  を  $\mathcal{CH}$  へ署名生成オラクルクエリとして送り,  $\sigma_{\text{SC},ksc}^{(i)}$  を受け取る.  $\mathcal{S}$  は  $\text{D.Encap}(K_{\text{SC},ksc}^{(i)}, m_{\text{SC},ksc}^{(i)} \| \sigma_{\text{SC},ksc}^{(i)}) =: (C_{D_{\text{SC},ksc}^{(i)}}^{(i)}, C_{\text{SC},ksc}^{(i)})$  を実行し,  $C_{\text{SC},ksc}^{(i)}$  を  $\mathcal{A}$  へ送り, リスト  $\mathcal{L}_{\text{SC},ksc} \rightarrow (m_{\text{SC},ksc}^{(i)}, C_{\text{SC},ksc}^{(i)}, S_s^{(i)}_{\text{SC},ksc}, S_d^{(i)}_{\text{SC},ksc}) = (m_{\text{SC},ksc}^{(i)}, (C_{K_{\text{SC},ksc}^{(i)}}^{(i)}, C_{D_{\text{SC},ksc}^{(i)}}^{(i)}), S_s^{(i)}_{\text{SC},ksc}, S_d^{(i)}_{\text{SC},ksc})$  を追加する.

**アンサインクリプション:**  $\mathcal{A}$  はサインクリプテキスト  $C = (C_K, C_D)$ , 署名者属性集合  $S_s$ , アクセス構造  $\mathbb{A}_r$  をクエリする.  $\mathcal{S}$  は  $\text{KK.KeyGen}(\text{PK}_{kk}, \text{MK}_{kk}, \mathbb{A}_r) \rightarrow \text{SK}_r$ ,  $\text{KK.Decap}(\text{PK}_{kk}, C_K, \text{SK}_r) =: \alpha$  を実行する. もし,  $\alpha \neq \perp$  ならば,  $K := \alpha$ ,  $\text{D.Decap}(K, C_D) =: \beta$  を実行する. もし,  $\beta \neq \perp$  ならば,  $m \| \sigma =: \beta$ ,  $\text{KS.Ver}(\text{PK}_{ks}, \sigma, m \| C_K, S_s) =: \gamma$  を実行する. もし,  $\gamma = 1$  ならば,  $\mathcal{S}$  は  $\mathcal{A}$  へ  $m$  を返す. もし,  $\alpha = \perp$  または  $\beta = \perp$  または  $\gamma = 0$  ならば,  $\perp$  を返す.

**Forgery Phase:**  $\mathcal{A}$  は  $C^* = (C_K^*, C_D^*), S_s^*, S_d^*$  を出力する. 安全性証明の便宜上, 各変数の表記を以下のように変更する.  $C_{\text{Frg},ksc}^* := C^*$ ,  $C_{K_{\text{Frg},ksc}^*}^* := C_K^*$ ,  $C_{D_{\text{Frg},ksc}^*}^* := C_D^*$ ,  $\mathbb{A}_{s_{\text{Frg},ksc}^*}^* := S_s^*$ ,  $S_{d_{\text{Frg},ksc}^*}^* := S_d^*$ .  $S_{d_{\text{Frg},ksc}^*}^* \in \mathbb{A}_r$  を満たす  $\mathbb{A}_r$  が  $N_{\text{Frg},ksc}$  個存在すると仮定し, それぞれを番号付けし,  $i$  番目を  $\mathbb{A}_r^{(i)}_{\text{Frg},ksc}$  ( $i = \{1, \dots, N_{\text{Frg},ksc}\}$ ) と表記する.  $\mathcal{S}$  は,  $\text{KK.KeyGen}(\text{PK}_{kk}, \text{MK}_{kk}, \mathbb{A}_r^{(i)}_{\text{Frg},ksc}) =: \text{SK}_r^{(i)}_{\text{Frg},ksc}$ ,  $\text{KK.Decap}(\text{PK}_{kk}, C_{K_{\text{Frg},ksc}^*}^*, \text{SK}_r^{(i)}_{\text{Frg},ksc}) =: \alpha^{(i)}$  を実行する. もし,  $\alpha^{(i)} \neq \perp$  ならば,  $\mathcal{S}$  は  $K_{\text{Frg},ksc}^{(i)} := \alpha^{(i)}$ ,  $\text{D.Decap}(K_{\text{Frg},ksc}^{(i)}, C_{D_{\text{Frg},ksc}^*}^*) =: \beta^{(i)}$  を実行する. もし,  $\beta^{(i)} \neq \perp$  ならば,  $\mathcal{S}$  は  $m_{\text{Frg},ksc}^{(i)} \| \sigma_{\text{Frg},ksc}^{(i)} := \beta^{(i)}$  を実行し,

$\text{KS.Ver}(\text{PK}_{ks}, \sigma_{\text{Frg},ksc}^{(i)}, m_{\text{Frg},ksc}^{(i)} \| C_{K\text{Frg},ksc}^*, S_{s\text{Frg},ksc}^*) = 1$  ならば,  $\gamma_{\text{Frg},ksc}^{(i)} = 1$  とする.  $\text{KS.Ver}(\text{PK}_{ks}, \sigma_{\text{Frg},ksc}^{(i)}, m_{\text{Frg},ksc}^{(i)} \| C_{K\text{Frg},ksc}^*, S_{s\text{Frg},ksc}^*) = 0$ , または  $\alpha^{(i)} = \perp$ , または  $\beta^{(i)} = \perp$  ならば,  $\gamma_{\text{Frg},ksc}^{(i)} = 0$  とする. ここまでの一連の処理を全ての  $i \in \{1, \dots, N_{\text{Frg},ksc}\}$  について実行し終わったら,  $\mathcal{S}$  は  $\gamma_{\text{Frg},ksc} := \gamma_{\text{Frg},ksc}^{(1)} \times \gamma_{\text{Frg},ksc}^{(2)} \times \dots \times \gamma_{\text{Frg},ksc}^{(N_{\text{Frg},ksc})}$  を実行する.  $\mathcal{S}$  は,  $j \stackrel{\text{U}}{\leftarrow} \{1, \dots, N_{\text{Frg},ksc}\}$  を実行し,  $\mathcal{CH}$  に対して AA-sEUF-CMA ゲームの Forgery Phase で,  $(m_{\text{Frg},ksc}^{(j)} \| C_{K\text{Frg},ksc}^*, \sigma_{\text{Frg},ksc}^{(j)}, S_{s\text{Frg},ksc}^*)$  を出力する.

$\mathcal{S}$  は  $\mathcal{A}$  に対して  $\Pi_{\text{KSC}}$  に関する AA-sEUF-CMA ゲームの挑戦者としての応答を完璧にシミュレートできている事はほぼ自明であるので, 詳細な説明は省略する.

$\mathcal{S}$  は  $\mathcal{A}$  が  $\Pi_{\text{KSC}}$  に関する AA-sEUF-CMA ゲームにおける禁止されたオラクルクエリを一度も行わなければ,  $\Pi_{\text{KS}}$  に関する AA-sEUF-CMA ゲームにおいて禁止されたオラクルクエリを行うことはないことも, ほぼ自明であるので, 詳細な説明は省略する.

従って,  $\Pi_{\text{KSC}}$  に関する AA-sEUF-CMA 安全性ゲームにおける  $\mathcal{A}$  の優位性はこれまでに定義した変数・記号を使って次式で定義できる.

$$\begin{aligned}
 \text{Adv}_{\Pi_{\text{KSC}}, \mathcal{A}}^{\text{AA-sEUF-CMA}} &= \Pr[ [\text{Disclose}_{\text{KSC}}(\text{PK}, C) = \text{Disclose}_{\text{KK}}(\text{PK}_{kk}, C_K^*) = S_d^*] \wedge [\gamma_{\text{Frg},ksc} = 1] \\
 &\quad \wedge [m_{\text{Frg},ksc}^{(1)} = \dots = m_{\text{Frg},ksc}^{(|S_{d\text{Frg},ksc}^*|)} =: m_{\text{Frg},ksc}^*] \\
 &\quad \wedge [(m_{\text{Frg},ksc}^*, (C_{K\text{Frg},ksc}^*, C_{D\text{Frg},ksc}^*), S_{s\text{Frg},ksc}^*, S_{d\text{Frg},ksc}^*) \notin \mathcal{L}_{\text{SC},ksc}] ] \\
 & \tag{7.13}
 \end{aligned}$$

ここで,  $\mathcal{CH}$  の動作を考えると, Query Phase の署名生成オラクルにおいて,  $\mathcal{S}$  が  $(m_{\text{SC},ksc}^{(i)} \| C_{K\text{SC},ksc}^{(i)}, A_s^{(i)} \text{SC},ksc, S_s^{(i)} \text{SC},ksc)$  をクエリすると,  $\mathcal{CH}$  は,  $\text{KS.KeyGen}(\text{PK}_{ks}, \text{MK}_{ks}, A_s^{(i)} \text{SC},ksc) =: \text{SK}_s^{(i)} \text{Sig},ks$  を実行する. 続けて,  $\text{KS.Sig}(\text{PK}_{ks}, m_{\text{SC},ksc}^{(i)} \| C_{K\text{SC},ksc}^{(i)}, \text{SK}_s^{(i)} \text{Sig},ks, S_s^{(i)} \text{SC},ksc) =: \sigma_{\text{Sig},ks}^{(i)}$  を実行する. そして, リスト  $\mathcal{L}_{\text{Sig},ks} \leftarrow (m_{\text{SC},ksc}^{(i)} \| C_{K\text{SC},ksc}^{(i)}, \sigma_{\text{Sig},ks}^{(i)}, S_s^{(i)} \text{SC},ksc)$  を追加する.

また, Forgery Phase における  $\mathcal{CH}$  の動作を考えると,  $\mathcal{CH}$  は  $\mathcal{S}$  より,  $(m_{\text{Frg},ksc}^{(j)} \| C_{K\text{Frg},ksc}^*, \sigma_{\text{Frg},ksc}^{(j)}, S_{s\text{Frg},ksc}^*) (j \in \{1, \dots, N_{\text{Frg},ksc}\})$  を出力される.  $\mathcal{CH}$  は署名検証のため,  $\text{KS.Ver}(\text{PK}_{ks}, \sigma_{\text{Frg},ksc}^{(j)}, m_{\text{Frg},ksc}^{(j)} \| C_{K\text{Frg},ksc}^*, S_{s\text{Frg},ksc}^*) =: \gamma_{\text{Frg},ksc} = 1 / 0$  を実行する. ここで,  $\text{KS.Ver}$  は確定的アルゴリズムであるため, 入力変数が同一であれば, 出力は必ず同一になるから, 次の等式が成り立つ.

$$\gamma_{\text{Frg},ks} = \gamma_{\text{Frg},ksc}^{(j)} \tag{7.14}$$

これまでに定義した変数, 記号を使って, KP-ABS 方式  $\Pi_{\text{KS}}$  に関する AA-sEUF-CMA 安全性ゲームにおける  $\mathcal{S}$  の優位性は次式で定義できる.

$$\text{Adv}_{\Pi_{\text{KS}}, \mathcal{S}}^{\text{AA-sEUF-CMA}} = \Pr[ [\gamma_{\text{Frg},ks} = 1] \wedge [(m_{\text{Frg},ksc}^{(j)} \| C_{K\text{Frg},ksc}^*, \sigma_{\text{Frg},ksc}^{(j)}, S_{s\text{Frg},ksc}^*) \notin \mathcal{L}_{\text{Sig},ks}] ] \tag{7.15}$$

ここで、安全性証明の便宜上、事象  $P_1, P_2, P_3, P_4, Q_1, Q_2$  を次の通りに定義する。

$$P_1 = [\text{Disclose}_{\text{KSC}}(\text{PK}, C) = \text{Disclose}_{\text{KK}}(\text{PK}_{kk}, C_K^*) = S_d^*] \quad (7.16)$$

$$P_2 = [\gamma_{\text{Frg}, k_{\text{sc}}} = 1] \quad (7.17)$$

$$P_3 = [m_{\text{Frg}, k_{\text{sc}}}^{(1)} = \cdots = m_{\text{Frg}, k_{\text{sc}}}^{(N_{\text{Frg}, k_{\text{sc}}})} =: m_{\text{Frg}, k_{\text{sc}}}^*] \quad (7.18)$$

$$P_4 = [(m_{\text{Frg}, k_{\text{sc}}}^*, (C_{K_{\text{Frg}, k_{\text{sc}}}^*}, C_{D_{\text{Frg}, k_{\text{sc}}}^*}), S_{s_{\text{Frg}, k_{\text{sc}}}^*}, S_{d_{\text{Frg}, k_{\text{sc}}}^*}) \notin \mathcal{L}_{\text{SC}, k_{\text{sc}}}] \quad (7.19)$$

$$Q_1 = [\gamma_{\text{Frg}, k_s} = 1] \quad (7.20)$$

$$Q_2 = [(m_{\text{Frg}, k_{\text{sc}}}^{(j)} \| C_{K_{\text{Frg}, k_{\text{sc}}}^*}, \sigma_{\text{Frg}, k_{\text{sc}}}^{(j)}, S_{s_{\text{Frg}, k_{\text{sc}}}^*}) \notin \mathcal{L}_{\text{Sig}, k_s}] \quad (7.21)$$

(以降で証明する) 補題 7.2.2.1 と補題 7.2.2.2 より、事象  $P := P_1 \wedge P_2 \wedge P_3 \wedge P_4$  が生じた場合、事象  $Q_1$  と事象  $Q_2$  はどちらも生起する。従って、次の不等式が成立する。

$$\text{Adv}_{\Pi_{\text{KS}}, \mathcal{S}}^{\text{AA-sEUUF-CMA}} \geq \text{Adv}_{\Pi_{\text{KSC}}, \mathcal{A}}^{\text{AA-sEUUF-CMA}} \quad (7.22)$$

$\text{Adv}_{\Pi_{\text{KSC}}, \mathcal{A}}^{\text{AA-sEUUF-CMA}}$  が無視できなくなるような  $\mathcal{A}$  が存在すると仮定すると、式 (7.22) より、 $\text{Adv}_{\Pi_{\text{KS}}, \mathcal{S}}^{\text{AA-sEUUF-CMA}}$  は無視できなくなり、これは  $\Pi_{\text{KS}}$  が AA-sEUUF-CMA 安全であるという事実に矛盾するので、その仮定は誤りである。従って、 $\Pi_{\text{KS}}$  が AA-sEUUF-CMA 安全ならば、いかなる PPTA  $\mathcal{A}$  に対しても、 $\text{Adv}_{\Pi_{\text{KSC}}, \mathcal{A}}^{\text{AA-sEUUF-CMA}}$  は無視できるほど小さい値になる。ゆえに、補題 7.2.2 は成立する。  $\square$

**補題 7.2.2.1.** 事象  $P = P_1 \wedge P_2 \wedge P_3 \wedge P_4$  が生じた場合、事象  $Q_1$  は生起する。つまり、 $\Pr[Q_1|P] = 1$  が成立する。

**補題 7.2.2.2.** 事象  $P = P_1 \wedge P_2 \wedge P_3 \wedge P_4$  が生じた場合、事象  $Q_2$  は生起する。つまり、 $\Pr[Q_2|P] = 1$  が成立する。

以降では、補題 7.2.2.1, 補題 7.2.2.2 の証明を示す。

**補題 7.2.2.1 の証明** 補題 5.2.2.1 の証明と全く同じ手順で証明可能であるので、詳細な記述は省略する。  $\square$

**補題 7.2.2.2 の証明** 証明の手順は補題 7.2.2.2 の証明と同様である。

事象  $P$  が生じた場合に事象  $Q_2$  は生起しないと仮定すると矛盾が生じることを示す。その仮定の下では、次の等式を満たすリスト  $\mathcal{L}_{\text{Sig}, k_s}$  内の  $k$  番目の要素  $(m_{\text{SC}, k_{\text{sc}}}^{(k)} \| C_{K_{\text{SC}, k_{\text{sc}}}^{(k)}}, \sigma_{\text{SC}, k_{\text{sc}}}^{(k)}, S_{s_{\text{SC}, k_{\text{sc}}}^{(k)}})$  が存在する。

$$(m_{\text{SC}, k_{\text{sc}}}^{(k)} \| C_{K_{\text{SC}, k_{\text{sc}}}^{(k)}}, \sigma_{\text{SC}, k_{\text{sc}}}^{(k)}, S_{s_{\text{SC}, k_{\text{sc}}}^{(k)}}) = (m_{\text{Frg}, k_{\text{sc}}}^{(j)} \| C_{K_{\text{Frg}, k_{\text{sc}}}^*}, \sigma_{\text{Frg}, k_{\text{sc}}}^{(j)}, S_{s_{\text{Frg}, k_{\text{sc}}}^*}) \quad (7.23)$$

式 (5.33) より以下の 4 つの等式が成立する。

$$m_{\text{SC}, k_{\text{sc}}}^{(k)} = m_{\text{Frg}, k_{\text{sc}}}^{(j)} \quad (7.24)$$

$$C_{K_{\text{SC}, k_{\text{sc}}}^{(k)}} = C_{K_{\text{Frg}, k_{\text{sc}}}^*} \quad (7.25)$$

$$\sigma_{\text{SC},ksc}^{(k)} = \sigma_{\text{Frg},ksc}^{(j)} \quad (7.26)$$

$$S_{s\text{SC},ksc}^{(k)} = S_{s\text{Frg},ksc}^* \quad (7.27)$$

$C_{K\text{SC},ksc}^{(k)}$  は  $\text{KK.Encap}(\text{PK}_{kk}, S_{d\text{SC},ksc}^{(k)})$  の実行により生成されている正当な鍵暗号文である事実と、 $\Pi_{\text{KK}}$  は復号者属性集合開示性を満たし、アルゴリズム  $\text{Disclose}_{\text{KK}}$  が存在することから、以下の等式が成立する。

$$\Pr[\text{Disclose}_{\text{KK}}(C_{K\text{SC},ksc}^{(k)} = S_{d\text{SC},ksc}^{(k)})] = 1 \quad (7.28)$$

一方で、事象  $P_1$  は生起すると仮定されているので、以下の等式が成立する。

$$\Pr[\text{Disclose}_{\text{KK}}(C_{K\text{Frg},ksc}^* = S_{s\text{Frg},ksc}^*)] = 1 \quad (7.29)$$

式 (7.25), (7.28), (7.29) より、以下の等式が成立する。

$$S_{d\text{SC},ksc}^{(k)} = S_{d\text{Frg},ksc}^* \quad (7.30)$$

$C_{K\text{SC},ksc}^{(k)}$  は、 $S$  が  $\text{KK.Encap}(\text{PK}_{kk}, S_{d\text{SC},ksc}^{(k)}) \rightarrow (K_{\text{SC},ksc}^{(k)}, C_{K\text{SC},ksc}^{(k)})$  を実行して生成したものなので、 $S_{d\text{SC},ksc}^{(k)} \in \mathbb{A}_r$  を満たす  $\mathbb{A}_r$  が  $N_{\text{SC},ksc}^{(k)}$  個存在するとし、それぞれを番号付けした上で  $i$  番目を  $\mathbb{A}_r^{(i)}_{\text{SC},ksc}$  と表記し、 $\text{KK.KeyGen}(\text{PK}_{kk}, \text{MK}_{kk}, \mathbb{A}_r^{(i)}_{\text{SC},ksc}) =: \text{SK}_r^{(i)}_{\text{SC},ksc}$  とすると、KP-ABKEM 方式  $\Pi_{\text{KK}}$  の正当性より、全ての  $i \in \{1, \dots, N_{\text{SC},ksc}^{(k)}\}$  について、以下の等式が成立する。

$$\Pr[\text{KK.Decap}(\text{PK}_{kk}, C_{K\text{SC},ksc}^{(k)}, \text{SK}_r^{(i)}_{\text{SC},ksc}) = K_{\text{SC},ksc}^{(k)}] = 1 \quad (7.31)$$

ここで、 $S_{d\text{Frg},ksc}^* \in \mathbb{A}_r$  を満たす  $\mathbb{A}_r$  が  $N_{\text{Frg},ksc}^*$  個存在するとし、それぞれを番号付けした上で  $i$  番目を  $\mathbb{A}_r^{(i)}_{\text{Frg},ksc}$  と表記する。式 (7.25), (7.30), (7.31) より、全ての  $i \in \{1, \dots, N_{\text{Frg},ksc}^*\}$  について、 $K_{\text{Frg},ksc}^*$  が存在し、以下の等式が成立する。

$$K_{\text{Frg},ksc}^{(i)} = K_{\text{SC},ksc}^{(k)} = K_{\text{Frg},ksc}^* \quad (7.32)$$

$i \in \{1, \dots, N_{\text{Frg},ksc}^*\}$  について、 $m_{\text{Frg},ksc}^{(i)}$  と  $\sigma_{\text{Frg},ksc}^{(i)}$  は、 $S$  が  $\text{D.Decap}(K_{\text{Frg},ksc}^{(i)}, C_{D\text{Frg},ksc}^*)$  を実行して生成したものである事実と、等式 (7.32) より、全ての  $i \in \{1, \dots, N_{\text{Frg},ksc}^*\}$  について、先述の  $\text{D.Decap}$  アルゴリズムへの入力変数は同じであり、 $\text{D.Decap}$  は確定的アルゴリズムであるので入力変数が同じであれば出力は同じになることから、全ての  $i \in \{1, \dots, N_{\text{Frg},ksc}^*\}$  について、 $m_{\text{Frg},ksc}^*$  と  $\sigma_{\text{Frg},ksc}^*$  が存在し、以下の等式が成立する。

$$m_{\text{Frg},ksc}^{(i)} \parallel \sigma_{\text{Frg},ksc}^{(i)} = m_{\text{Frg},ksc}^* \parallel \sigma_{\text{Frg},ksc}^* \quad (7.33)$$

式 (7.24), (7.26), (7.33) より、以下の等式が成立する。

$$m_{\text{SC},ksc}^{(k)} = m_{\text{Frg},ksc}^{(j)} = m_{\text{Frg},ksc}^* \quad (7.34)$$

$$\sigma_{\text{SC},ksc}^{(k)} = \sigma_{\text{Frg},ksc}^{(j)} = \sigma_{\text{Frg},ksc}^* \quad (7.35)$$

$i \in \{1, \dots, N^*_{\text{Frg},ksc}\}$  について,  $m_{\text{Frg},ksc}^{(i)}$  と  $\sigma_{\text{Frg},ksc}^{(i)}$  は,  $\mathcal{S}$  が  $\text{D.Decap}(K_{\text{Frg},ksc}^{(i)}, C_{\text{DFrg},ksc}^*)$  を実行して生成したものである事実と, 式 (7.32), (7.33) より, 以下の等式が成立する.

$$\Pr[\text{D.Decap}(K_{\text{Frg},ksc}^*, C_{\text{DFrg},ksc}^*) = m_{\text{Frg},ksc}^* \parallel \sigma_{\text{Frg},ksc}^*] = 1 \quad (7.36)$$

$C_{\text{D SC},ksc}^{(k)}$  は,  $\mathcal{S}$  が  $\text{D.Encap}(K_{\text{SC},ksc}^{(k)}, m_{\text{SC},ksc}^{(k)} \parallel \sigma_{\text{SC},ksc}^{(k)})$  を実行して生成したものである事実と, DEM 方式  $\Pi_{\text{D}}$  の正当性より, 以下の等式が成立する.

$$\Pr[\text{D.Decap}(K_{\text{SC},ksc}^{(k)}, C_{\text{D SC},ksc}^{(k)}) = m_{\text{SC},ksc}^{(k)} \parallel \sigma_{\text{SC},ksc}^{(k)}] = 1 \quad (7.37)$$

式 (7.32), (7.34), (7.35), (7.36), (7.37),  $\Pi_{\text{D}}$  の一対一対応性より, 以下の等式が成立する.

$$C_{\text{D SC},ksc}^{(k)} = C_{\text{DFrg},ksc}^* \quad (7.38)$$

式 (7.25), (7.27), (7.30), (7.34), (7.38) より, 以下の等式が成立する.

$$\begin{aligned} & (m_{\text{SC},ksc}^{(k)}, (C_{\text{K SC},ksc}^{(k)}, C_{\text{D SC},ksc}^{(k)}), S_s^{(k)}, S_d^{(k)}) \\ &= (m_{\text{Frg},ksc}^*, (C_{\text{KFrg},ksc}^*, C_{\text{DFrg},ksc}^*), S_s^*, S_d^*) \end{aligned} \quad (7.39)$$

等式 (7.39) は事象  $P$  が生じた場合という条件に対する矛盾を表しているので, 事象  $Q_2$  が生じないという仮定が誤りである. ゆえに, 事象  $P$  が生じた場合, 事象  $Q_2$  は必ず生起する. ゆえに, 補題 7.2.2.2 は成立する.  $\square$

**定理 7.3 の証明** 証明の手順は定理 5.3 の証明と同様である.

任意の  $k$ , 任意の  $\mathcal{U}_s$ , 任意の  $\mathcal{U}_r$  に対し,  $(\text{PK}_{ks}, \text{MK}_{ks}) \leftarrow \text{KS.Setup}(1^k, \mathcal{U}_s)$  を実行し,  $(\text{PK}_{kk}, \text{MK}_{kk}) \leftarrow \text{KK.Setup}(1^k, \mathcal{U}_r)$  を実行する. さらに, 任意の  $\mathbb{A}_s$ , 任意の  $\mathbb{A}'_s$  に対し,  $\text{SK}_s \leftarrow \text{KS.KeyGen}(\text{PK}_{ks}, \text{MK}_{ks}, \mathbb{A}_s)$ ,  $\text{SK}'_s \leftarrow \text{KS.KeyGen}(\text{PK}_{ks}, \text{MK}_{ks}, \mathbb{A}'_s)$  を実行する. そして, 任意の  $m \in \mathcal{M}$ , 任意の  $S_s$  (s.t.  $S_s \in (2^{\mathcal{U}_s} - \{\phi\}) \wedge S_s \in \mathbb{A}_s \wedge S_s \in \mathbb{A}'_s$ ), 任意の  $S_d \in (2^{\mathcal{U}_r} - \{\phi\})$  に対して, 以下の一連の処理を実行する.  $(K, C_K) \leftarrow \text{KK.Encap}(\text{PK}_{kk}, S_d)$ ,  $\sigma \leftarrow \text{KS.Sig}(\text{PK}_{ks}, m \parallel C_K, \text{SK}_s, S_s)$ ,  $\sigma' \leftarrow \text{KS.Sig}(\text{PK}_{ks}, m \parallel C_K, \text{SK}'_s, S_s)$ ,  $C_D \leftarrow \text{D.Encap}(K, m \parallel \sigma)$ ,  $C'_D \leftarrow \text{D.Encap}(K, m \parallel \sigma')$ ,  $C := (C_K, C_D)$ ,  $C' := (C_K, C'_D)$ .

ここで,  $\text{KP-ABS}\Pi_{\text{KS}}$  は完全匿名だから,  $\sigma$  の確率分布と,  $\sigma'$  の確率分布は同一である.

そして,  $\sigma$  の確率分布と,  $\sigma'$  の確率分布は同一であれば,  $m \parallel \sigma$  の確率分布と,  $m \parallel \sigma'$  の確率分布は同一である.

さらに,  $m \parallel \sigma$  の確率分布と,  $m \parallel \sigma'$  の確率分布は同一であれば,  $C_D$  の確率分布と,  $C'_D$  の確率分布は同一である.

最後に,  $C_D$  の確率分布と,  $C'_D$  の確率分布は同一であれば,  $C$  の確率分布と,  $C'$  の確率分布は同一である.

従って,  $\text{KP-ABSC}$  の一般的構成  $\Pi_{\text{KSC}}$  は, 完全匿名である. ゆえに, 定理 7.3 が成立する.  $\square$

## Chapter 8 議論

### 8.1 本研究の意義

#### 8.1.1 鍵カプセル化メカニズムの一般的構成（4章,6章）の意義

本項では、4章の暗号文ポリシー型属性ベース鍵カプセル化メカニズム (CP-ABKEM) の一般的構成の成果、及び6章の鍵ポリシー型属性ベース鍵カプセル化メカニズム (KP-ABKEM) の一般的構成の成果に関して、具体的にどのような意義があるかについて述べる。なお、以下では特に、CP-ABKEM の一般的構成の成果の意義について述べるが、KP-ABKEM の一般的構成の意義についても全く同様の議論が成り立つので、KP-ABKEM の一般的構成の意義についての説明は省略する。

4章で提案したCP-ABKEMの一般的構成法は、構成法自体は自明な構成法である。具体的には、鍵カプセル化を実行するにあたっては、送受信者間で共有するセッション鍵  $K$  を鍵空間  $\mathcal{K}$  からランダムに取り出し、セッション鍵  $K$  をCP-ABE方式の暗号化アルゴリズムで暗号化するという、極めて単純な構成法である。構成法は自明であるが、安全性証明は自明ではない。そして、著者の知る限りでは、本成果で完成させた安全性証明を過去に完成させた既存研究はない。従って、それが本成果の意義の一つである。

また、この種の研究分野では、CP-ABEに関しては、過去から現在に至るまで非常に活発に研究されており、具体的構成法も多数発表されているが、その一方でCP-ABKEMはほとんど活発な研究は行われておらず、既存方式は極めて少数である。そして、今後もこの傾向は続くと思われる。さらに、CP-ABEにとっての最強の安全性とみなされているAP-IND-CCA安全性を達成可能なCP-ABE方式は過去に多数提案されているが、著者の知る限りでは、CP-ABKEMにとっての最強の安全性とみなされているAP-IND-CCA安全性を達成可能なCP-ABKEM方式は過去に一つも提案されていないという現状もある。従って、本成果が実際上、既存の及び今後発表されるであろう多数のAP-IND-CCA安全なCP-ABEから、多数のAP-IND-CCA安全なCP-ABKEMが具体的に構成可能であることを証明した点を考慮すると、本成果は有意義であると考えられる。

また、本稿の5章で示した、AP-IND-CCA安全性を達成可能な暗号文ポリシー型属性ベース Signcryption(CP-ABSC)の一般的構成では、AP-IND-CCA安全かつ復号者アクセス構造開示的なCP-ABKEMを構成要素として用いている。従って、CP-ABKEMの一般的構成という本成果は、5章のCP-ABSCの一般的構成においても、有用であると言える。

## 8.1.2 暗号文ポリシー型属性ベース Signcryption の一般的構成 (5章) の意義

本項では、5章の成果である暗号文ポリシー型属性ベース Signcryption(CP-ABSC)の一般的構成に関して、具体的にどのような意義があるかについて記述する。

### 具体的構成を考えた場合の意義

表 8.1 に AP-IND-CCA 安全性及び AP-sEUF-CMA 安全性及び完全匿名性を達成可能な CP-ABSC 方式の具体的構成を列挙する。具体的には、表 8.1 には、Pandit ら [4][5] が提案した具体的構成、Nandi ら [9] が提案した一般的構成を基に作れる具体的構成の中で“unbounded”であるもの、そして本研究で提案した一般的構成(図 5.1)に基づく2つの具体的構成を列挙している。

表 8.1 中の略語と略す前の単語の対応は次の通りである。MAS=Monotone Access Structure(単調アクセス構造), NMAS=Non-Monotone Access Structure(非単調アクセス構造), DSG=Decisional Sub-Group, EDHE4D=Expanded Diffie-Hellman Exponent 4-Dual, EDHE3=Expanded Diffie-Hellman Exponent 3, DLIN=Decisional Linear, CS=Combined Setup. 以下、単調アクセス構造と非単調アクセス構造の違い、large universe と small universe の違い、“Bounded”と“Unbounded”の違いについて、簡単に説明する。

まず、単調アクセス構造は直感的には属性を変数として AND 演算子, OR 演算子, Threshold 演算子を用いて表せる論理式と等価であるアクセス構造を表し、非単調アクセス構造は直感的には属性を変数として AND 演算子, OR 演算子, Threshold 演算子に加えて NOT 演算子を用いて表せる論理式と等価であるアクセス構造を表す。単調アクセス構造の方が、非単調アクセス構造よりも、良い(望ましい)性質である。

また、large universe は属性の全体集合のサイズが指数的に大きい、つまりセキュリティパラメータ  $k$  に関する指数関数で表されることを表し、一方 small universe は属性の全体集合のサイズがセキュリティパラメータ  $k$  に関する多項式関数で表されることを表す。large universe の方が、small universe よりも、良い(望ましい)性質である。

そして、“(Completely)Unbounded”は、[24][25]を参考にして述べると、「属性集合  $(S_s, S_r)$  のサイズが制約されず、かつアクセス構造  $(A_d, A_s)$  のサイズが制約されず、かつアクセス構造  $(A_s, A_d)$  と等価な属性を変数とする論理式において同一属性の使用回数に制限がかからない」性質を表す。対して、“Bounded”は先の3つの条件のいずれかを満たさない場合を指す。“Unbounded”の方が、“Bounded”よりも、良い(望ましい)性質である。

本研究の具体的構成 1 は、アクセス構造(正確には、復号者アクセス構造と署名者アクセス構造の両方)が非単調であり、かつ最強の安全性を達成可能な、最初の CP-ABSC 方式の具体的構成である。また、(証明が未完成の  $\Pi_{SS}$  の署名者アクセス構造開示性を除いて、) 仮定は全体的に比較的弱いと考えられる。

本研究の具体的構成 2 は、unbounded であり、かつ最強の安全性を達成可能である。そして、既存の unbounded であり、かつ最強の安全性を達成可能な具体的構成(Pandit らの構成, Nandi らの構成)と比べると、(証明が未完成の  $\Pi_{SS}$  の署名者アクセス構造開示性を除いて、) 仮定が全体的により弱い。本研究の具体的構成 2 以外の、unbounded

である構成2つはいずれも、Decisional Sub-Group 仮定という強い仮定が3種類必要であり、さらに Nandi らの構成に関しては、その Decisional Sub-Group 仮定3種に加えて、同様に強い仮定である EDHE4-Dual 仮定 [24][25] と EDHE3 仮定 [24][25] の両方が必要である。対して、本研究の具体的構成2に関しては、DLIN 仮定は比較的弱い仮定と考えられており、他の仮定もいずれも比較的弱い仮定と考えられる。

従って、本研究の CP-ABSC 方式の一般的構成を基に、以下の性質を満たす初めての CP-ABSC 方式を具体的に構成できる。

- AP-IND-CCA 安全, かつ AP-sEUF-CMA 安全, かつ完全匿名, かつアクセス構造が非単調, かつ“large universe”, かつ DLIN 仮定などの比較的弱い仮定に基づく, CP-ABSC 方式.
- AP-IND-CCA 安全, かつ AP-sEUF-CMA 安全, かつ完全匿名, かつ“large universe”, かつ“unbounded”, かつ DLIN 仮定などの比較的弱い仮定に基づく, CP-ABSC 方式.

実際には、本研究の具体的構成1,2は、Pandit らの構成及び Nandi らの構成と比べると、欠点が2つある。

欠点の一つ目は、Pandit らの構成及び Nandi らの構成は Combined Setup 型であるのに対して、本研究の具体的構成は1,2はいずれも非 Combined Setup 型であることによって、本研究の具体的構成1,2を用いる場合は、Pandit らの構成及び Nandi らの構成を用いる場合と比べると、一般のユーザが管理しなければいけない秘密鍵及び公開鍵の個数が倍になるという欠点である。この欠点に関しては、詳しくは8.2.1節で述べる。

欠点の二つ目は、本研究の具体的構成1,2が採用している SP-ABS 方式  $\Pi_{SS}$ (具体的には、Okamoto らが提案した SP-ABS[28][29] と Maji らが提案した SP-ABS[27])は、いずれも本研究で独自に新たに定義した署名者アクセス構造開示性と命名した性質を満たすことがまだ証明できていない点である。この欠点に関しては、詳しくは8.2.2節で述べる。

### 一般的構成自体の意義

表8.2を使って AP-IND-CCA 安全性, AP-sEUF-CMA 安全性, 完全匿名性を達成可能な CP-ABSC 方式の一般的構成の比較を行う。具体的には、当該安全性を達成可能な一般的構成は、Nandi ら [9] の構成と本研究の構成 (図5.1) の二つのみである。

Nandi らの構成は、比較的強い困難性仮定である Decisional Sub-Group 仮定が3種類、少なくとも必要である。また、Nandi らの構成は、構成要素として、“特定の安全性・条件を満たす”Pair Encoding[10][11]を用いており、現時点までに提案されている Pair Encoding の多くが“特定の安全性・条件を満たす”ことを証明するために、表8.1で Nandi らの具体的構成が採用した Pair Encoding が EDHE4D 仮定や EDHE3 仮定という強い仮定を必要としたのと同様に、それらと同程度に強い仮定を必要とする。DSG 仮定3種類と他の強い仮定のうちの一つでも仮定が成立しないことが証明されたら、Nandi らの構成は安全性が保障されなくなる。

それに対して、本研究の構成は安全性を構成要素 (CP-ABKEM, SP-ABS, DEM) の安全性に帰着させており、表 8.1 で DLIN 仮定という比較的弱い仮定に基づく具体的構成を例示したように、弱い仮定に基づく方式を構成要素として採用することで弱い仮定に基づく方式を構成することが可能である。この研究分野は現在も活発な研究が行われている分野であるので、今後 DLIN 仮定と同程度に弱いもしくは DLIN 仮定よりも弱い仮定に基づく、CP-ABE 方式、SP-ABS 方式が多数発表される可能性が十分にある。ならば、それらを本研究の構成要素として採用することで、DLIN 仮定と同程度もしくはそれ以上に弱い仮定に基づく、表 8.1 の“本研究の構成法に基づく具体的構成法 1,2”以外の様々な特長（例えば、計算コストが小さい(一定)、秘密鍵(暗号文)サイズが小さい(一定)、など)を持った CP-ABSC 方式を多数具体的に構成できるようになる可能性が十分にあると言える。従って、本研究の CP-ABSC の一般的構成は十分有意義であると考えられる。

本研究の一般的構成は、Nandi らの一般的構成と比較すると、前パラグラフで具体的構成を考えた場合と同様の欠点が 2 つある。欠点は、本研究の構成がユーザが管理すべき鍵の個数が倍になることと、 $\Pi_{SS}$  の署名者アクセス構造衝突困難性の証明法が発見できていないことである。これらの欠点については、それぞれ 8.2.1, 8.2.2 で述べる。

方式	アクセス構造	Universe	Bounded or Unbounded	仮定	CS?
Pandit ら [4][5] の具体的構成	MAS	large	bounded	衝突困難なハッシュ関数, 秘匿性 (hiding property) を持つコミットメント, sEUF-CMA 安全なワンタイム署名, 3 種類の DSG 仮定.	Yes
Nandi ら [9] の構成に基づく具体的構成 1 (Pair Encoding:[24][25])	MAS	large	unbounded	衝突困難なハッシュ関数, 秘匿性 (hiding property) を持つコミットメント, sEUF-CMA 安全なワンタイム署名, EDHE4D 仮定 [24][25], EDHE3 仮定 [24][25], 3 種類の DSG 仮定.	Yes
本研究の構成 (図 5.1) に基づく具体的構成 1 ( $(\Pi_{CK}(\Pi_{CE}):[22][23], \Pi_{SS}):[28][29]$ )	NMAS	large	bounded	衝突困難なハッシュ関数, sEUF-CMA 安全なワンタイム署名, DLIN 仮定, IND-CCA 安全かつ一対一対応な DEM, $\Pi_{SS}$ が署名者アクセス構造衝突困難であること.	No
本研究の構成 (図 5.1) に基づく具体的構成 2 ( $(\Pi_{CK}(\Pi_{CE}):[22][23], \Pi_{SS}):[27]$ )	NMAS( $A_d$ ) MAS( $A_s$ )	large	unbounded	sEUF-CMA 安全なワンタイム署名, DLIN 仮定, IND-CCA 安全かつ一対一対応な DEM, $\Pi_{SS}$ が署名者アクセス構造衝突困難であること.	No

表 8.1: AP-IND-CCA 安全, かつ AP-sEUF-CMA 安全, かつ完全匿名な CP-ABSC 方式の具体的構成の比較

方式	仮定	CS?
Nandi ら [9]	衝突困難なハッシュ関数, 秘匿性 (hiding property) を持つコミットメント, sEUF-CMA 安全なワンタイム署名, (特定の安全性・条件を満たす) Pair Encoding, 3種類の Decisional Sub-Group 仮定.	Yes
本研究 (5 章)	AP-IND-CCA 安全かつ復号者アクセス構造開示的な CP-ABKEM(CP-ABE), AP-sEUF-CMA 安全かつ署名者アクセス構造衝突困難な SP-ABS, IND-CCA 安全かつ一対一対応な DEM.	No

表 8.2: AP-IND-CCA 安全, かつ AP-sEUF-CMA 安全, かつ完全匿名な CP-ABSC 方式の一般的構成の比較

### 8.1.3 鍵ポリシー型属性ベース Signcryption の一般的構成 (7 章) の意義

本項では, 7 章の鍵ポリシー型属性ベース Signcryption(KP-ABSC) の一般的構成の成果に関して, 具体的にどのような意義があるかについて記述する.

#### 具体的構成を考えた場合の意義

前項 (8.1.2) のパラグラフ “具体的構成を考えた場合の意義” では, 本研究の CP-ABSC 方式の一般的構成から, 既存方式に対して優位な性質を持った方式を具体的に構成することができることを説明した. 具体的には, 表 8.1 の “本研究の構成 (図 5.1) に基づく具体的構成 1” は, 最強の安全性を達成可能であり, かつアクセス構造が非単調であり, かつ仮定が比較的弱いという特長を持った初めての方式であった. また, 表 8.1 の “本研究の構成 (図 5.1) に基づく具体的構成 2” は, 最強の安全性を達成可能であり, かつ “large universe” であり, かつ “unbounded” であり, かつ仮定が比較的弱いという特長を持った初めての方式であった.

それに対して, KP-ABSC の場合, 本研究の構成 (図 7.1) からは, 既存の KP-ABE(KP-ABKEM), KP-ABS 方式を用いて, 「既存方式に対して優位な性質 (特長) をを持った方式」を具体的に構成することは, 現状不可能である. それは, KP-ABS に関する研究がこれまで活発に行われていなかったため, これまでに提案された KP-ABS 方式が少ないことに起因する.

KP-ABE に関しては, AA-IND-CCA 安全であり, かつアクセス構造が非単調であり, かつ DLIN 仮定などの弱い仮定に基づく, 具体的な KP-ABE 方式が Okamoto らによる [22][23] などによって提案されている. しかし, KP-ABS に関しては, AA-sEUF-CMA 安全であり, かつ完全匿名であり, かつアクセス構造が非単調であり, かつ DLIN 仮定などの弱い仮定に基づく, 具体的な KP-ABS 方式がまだ提案されていない. 従って, 表 8.1 の “本研究の構成 (図 5.1) に基づく具体的構成 1” の “KP-ABSC 版” は作れない.

同様に, AA-IND-CCA 安全であり, かつ “large universe” であり, かつ “unbounded” であり, かつ仮定が比較的弱い, 具体的な KP-ABE 方式は Okamoto らによる [22][23] などによって提案されている. しかし, AA-sEUF-CMA 安全であり, かつ完全匿名であり, かつ “large universe” であり, かつ “unbounded” であり, かつ仮定が比較的弱い,

具体的な KP-ABS 方式はまだ提案されていない。よって、表 8.1 の“本研究の構成(図 5.1)に基づく具体的な構成 2”の“KP-ABSC 版”は作れない。

以上の理由から、本研究の KP-ABSC 方式の一般的構成(図 7.1)からは、既存の KP-ABE(KP-ABKEM), KP-ABS 方式を用いて、「既存方式に対して優位な性質(特長)を持った方式」を具体的に構成することは、現時点では不可能である。

### 一般的構成自体の意義

表 8.3 に AA-IND-CCA 安全性, AA-sEUF-CMA 安全性, 完全匿名性を達成可能な KP-ABSC 方式の一般的構成を列挙する。具体的には、当該安全性を達成可能であることが証明されている KP-ABSC 方式の一般的構成は、Nandi ら [9] の構成と本研究の構成(7章)の二つのみである。

本研究の KP-ABSC 方式の一般的構成(7章)自体にどのような意義があるか、そして Nandi らの [9] の構成と比較したときにどのような優位性があるか、については、CP-ABSC の場合と同様の議論が成り立つ。つまり、「Nandi らの構成は DSG 仮定などの強い仮定が少なくとも必要であり、対して本研究の構成は今後 DLIN 仮定、もしくは DLIN 仮定と同程度に弱い仮定、もしくはそれ以上に弱い仮定に基づく、KP-ABE(KP-ABKEM), KP-ABS 方式が多数提案された場合に、それらを構成要素として採用することで“弱い”仮定に基づき、かつ様々な特長(効率が良い、など)を持った KP-ABSC 方式を多数具体的に構成できるようになる」といった議論が成り立つ。

方式	仮定	CS?
Nandi ら [9]	衝突困難なハッシュ関数, 秘匿性 (hiding property) を持つコミットメント, sEUF-CMA 安全なワンタイム署名, (特定の安全性・条件を満たす) Pair Encoding, 3 種類の Decisional Sub-Group 仮定.	Yes
本研究 (7 章)	AA-IND-CCA 安全かつ復号者属性集合開示的な KP-ABKEM(KP-ABE), AA-sEUF-CMA 安全かつ署名者属性集合衝突困難な KP-ABS, IND-CCA 安全かつ一対一対応な DEM.	No

表 8.3: AA-IND-CCA 安全, かつ AA-sEUF-CMA 安全, かつ完全匿名な KP-ABSC 方式の一般的構成の比較

## 8.2 本研究の課題

### 8.2.1 非 Combined-Setup 型属性ベース Signcryption について

**Combined Setup 型 ABSC と非 Combined Setup 型 ABSC** 本研究で提案した暗号文ポリシー型属性ベース Signcryption の一般的構成法(図 5.1)と鍵ポリシー型属性ベース Signcryption の一般的構成法(図 7.1)はいずれも、非 Combined Setup 型である。対して、CP-ABSC に関しては、表 3.1 または表 8.1 または表 8.2 にあるように、Pandit ら [4][5] の構成法, Nandi ら [9] の構成法は、Combined Setup 型である。また、KP-ABSC に関

しては、表 3.1 または表 8.3 にあるように、Nandi ら [9] の構成法は、Combined Setup 型である。

Combined Setup 型の ABSC は、シンタックスにおいて、システム公開鍵  $PK$ 、マスター秘密鍵  $MK$ 、秘密鍵生成アルゴリズムがそれぞれ一つしかない。対して、非 Combined Setup 型の CP-ABSC (resp. KP-ABSC) の場合、本稿の定義 (2.8 節 (resp. 2.9 節)) にあるように、システム公開鍵は  $PK_{ss}, PK_{ck}$  (resp.  $PK_{ks}, PK_{kk}$ ) の二つが必要、マスター秘密鍵は  $MK_{ss}, MK_{ck}$  (resp.  $MK_{ks}, MK_{kk}$ ) の二つが必要、そして受信者用秘密鍵生成アルゴリズムと送信者用秘密鍵生成アルゴリズムの二つの秘密鍵生成アルゴリズムが必要である。シンタックスにおける秘密鍵生成アルゴリズムの個数の違いは、ユーザが管理すべき秘密鍵の数に関連する。つまり、Combined Setup 型では、ユーザが管理すべき秘密鍵は一つだけで、この秘密鍵でサインクリプションとアンサインクリプションの両方を実行できる。対して、非 Combined Setup 型では、ユーザは送信者用秘密鍵  $SK_s$  と、受信者用秘密鍵  $SK_r$  の二つの秘密鍵を管理し、サインクリプションを行う際には  $SK_s$  を、アンサインクリプションを行う際には  $SK_r$  を用いるというように、使い分けが必要になる。

**非 Combined Setup 型 ABSC の欠点について** 非 Combined Setup 型の ABSC の実用上の欠点は、ユーザが管理すべき鍵の個数が倍になる点である。Combined Setup 型の ABSC を用いる場合は、一般のユーザが管理すべき鍵は、システム公開鍵  $PK$  と、自身の秘密鍵  $SK$  である。対して、非 Combined Setup 型の CP-ABSC を用いる場合は、一般のユーザが管理すべき鍵は、システム公開鍵 2 つ  $PK_{ss}, PK_{ck}$  と、自身の秘密鍵 2 つ  $SK_s, SK_r$  である。非 Combined Setup 型の KP-ABSC についても、同様である。従って、本研究で提案した CP-ABSC 及び KP-ABSC の一般的構成は、既存の Combined Setup 型の関連方式と比較して、一般的なユーザが管理すべき鍵の個数が倍になるという欠点を持っている。

**本研究の ABSC の一般的構成の Combined Setup 型への変換は可能か？** 本研究で提案した CP-ABSC の一般的構成の Combined Setup 版を作ることは可能であるか考える。(以降は CP-ABSC についてのみ論じるが、KP-ABSC に関しては同様の議論が成り立つことを理由に省略する。) つまり、AP-IND-CCA 安全性、AP-sEUF-CMA 安全性、完全匿名性を達成可能であり、かつ安全性を構成要素の CP-ABKEM, SP-ABS の安全性に直接的に帰着させ、かつ Combined Setup 型である、CP-ABSC の一般的構成である。以降これを「Combined Setup 型版  $\Pi_{CS}$ 」と表記する。また、非 Combined Setup 型である本研究の CP-ABSC の一般的構成は、単純に「 $\Pi_{CS}$ 」と表記する。

$\Pi_{CS}$  が Combined Setup 型版  $\Pi_{CS}$  と比較して、ユーザが管理すべき鍵の個数が増えるという欠点は確かにあるが、Combined Setup 型版  $\Pi_{CS}$  が  $\Pi_{CS}$  より“真に優れているか”と言う質問に対しては、肯定はできない。

Combined Setup 型版  $\Pi_{CS}$  は、構成要素として使用できる CP-ABKEM(CP-ABE) と SP-ABS に強い制限がかかるという問題がある。具体的に説明すると、Combined Setup 型版  $\Pi_{CS}$  の構成要素である CP-ABKEM(CP-ABE) と SP-ABS は、セットアップアルゴリ

ズムと秘密鍵生成アルゴリズムが完全に同一でなければならない。適当に構成した CP-ABKEM(CP-ABE) 方式と SP-ABS 方式のセットアップアルゴリズムと秘密鍵生成アルゴリズムが同一になることはあり得ないので、通常はそのような CP-ABKEM(CP-ABE) 方式と SP-ABS 方式を意図的に作ろうとしない限りは作れない。つまり、Combined Setup 型版  $\Pi_{CS}$  は、非 Combined Setup 型の  $\Pi_{CS}$  と比べると、構成要素として使用できる CP-ABKEM(CP-ABE) 方式、SP-ABS 方式が極めて限定され、使用できる方式の個数が非常に少なくなる。従って、Combined Setup 型版  $\Pi_{CS}$  は、非 Combined Setup 型の  $\Pi_{CS}$  と比べると、その一般的構成を基にして作れる具体的構成の数が非常に少なくなるという欠点がある。この欠点は、危殆化のリスクが増えるという意味で、実用的に問題である。

以上の理由から、Combined Setup 型版  $\Pi_{CS}$  と  $\Pi_{CS}$  には、どちらにも利点と欠点があるため、片方がもう片方に対して“真に優れている(劣っている)”とは言えない。

Combined Setup 型版  $\Pi_{CS}$  が非 Combined Setup 型である  $\Pi_{CS}$  よりも、“真に劣っている”わけではないので、Combined Setup 型版  $\Pi_{CS}$  を実現するための方法を検討することには十分意味がある。結論から言うと、直感的には Combined Setup 型版  $\Pi_{CS}$  は本研究の結果の大部分を参考にして比較的容易に実現できそうである。Combined Setup 型版  $\Pi_{CS}$  を実現する上で解決すべき主な課題は、以下の三つである。

**課題 1:** Combined Setup 型の CP-ABSC のシンタックス及び正当性を厳密に定義する。

**課題 2:** Combined Setup 型の CP-ABSC の安全性定義 (AP-IND-CCA, AP-sEUF-CMA, 完全匿名性) を厳密に定義する。

**課題 3:** 安全性証明

課題 1,2 が極めて容易であることは自明であり、問題は課題 3 の安全性証明である。

課題 3 の安全性証明に関しては、直感的には本研究の安全性証明の大部分をほぼそのまま流用できそうである。具体的には、定理 5.1(AP-IND-CCA) に関しては、補題 5.1.2 以外は、証明を“Combined Setup 型”仕様に変更するだけでうまくいくと考えられる。補題 5.1.2 の証明に関しては、 $\mathcal{A}$  がサインクリプションオラクルに発行するクエリ内の  $S_s$  が  $S_s \in \mathbb{A}_d^*$  を満たす場合、 $\mathcal{S}$  は  $S_s$  を  $\mathcal{CH}$  へ秘密鍵生成オラクルとして送ることができない(禁止されている)ため、 $\mathcal{A}$  からのオラクルクエリに対して  $\mathcal{S}$  は正しい返答ができないという問題が生じる。しかし、直感的には  $\mathcal{A}$  がそのようなクエリを発行することによって、Combined Setup 型の  $\Pi_{CS}$  に関する  $\text{Game}_1$  または  $\text{Game}_2$  に勝利しやすくなる可能性は極めて小さいと考えられるので、そのようなクエリに対して常に  $\perp$  を返すようなゲームを新たに定義して加えることによって、証明は通る可能性が高そうである。Combined Setup 型の  $\Pi_{CS}$  の AP-sEUF-CMA 安全性の証明も、定理 5.2 の証明を大部分流用することでうまくいくのではないかという直感を抱いている。

以上述べた通り、Combined Setup 型版  $\Pi_{CS}$  に関しては、本研究の結果を基にして比較的容易に実現できるのではないかという直感を抱いている。Combined Setup 型の  $\Pi_{CS}$  の厳密な安全性証明は今後の課題である。

## 8.2.2 SP-ABS(resp. KP-ABS)の署名者アクセス構造開示性(resp. 署名者属性集合開示性)について

本研究のCP-ABSC方式の一般的構成は、安全性を証明するための仮定として、SP-ABSの署名者アクセス構造開示性という性質を利用している。この性質は、本研究で独自に定義したものであり、既存のSP-ABS方式がこの性質を満たすことの証明はまだ完成していない。既存のSP-ABS方式がこの性質を満たさない場合、実用上の特定の問題が生じると考えられるので、直感的には既存の方式がこの性質を満たす可能性は高そうである。以下、SP-ABS方式がこの性質を満たさない場合に生じる「実用上の特定の問題」について、解説する。(なお、以降はSP-ABSの署名者アクセス構造開示性についてのみ取り上げ、KP-ABSの署名者属性集合開示性に関しては同様の議論が成り立つことを理由に説明を割愛する。)

SP-ABS方式がこの性質を満たさない場合、 $SS.Ver(PK_{ss}, \sigma^*, m^*, A_s^*) = 1$  及び  $SS.Ver(PK_{ss}, \sigma^*, m^*, A'_s) = 1$  を満たす2つの異なる署名者アクセス構造  $A_s^*, A'_s$  (s.t.  $A_s^* \neq A'_s$ ) が存在する。上の条件を満たすような2つの異なる署名者アクセス構造  $A_s^*, A'_s$  の存在が実用上問題を生じさせる可能性があることは以下の「内部告発」の簡単な例を用いることで理解できる。

例えば、“経理部”の“平社員”であるアリスは、所属する経理部で行われた重大な社内不正の事実をある日知る。そこで、アリスはこの事実を社内の内部告発対応部署に知らせようとする。アリスは、署名作成者が確かに経理部に所属する人物であることを、署名検証者に対して証明したい。アリスは、平文を  $m^* =$  「経理部で社内不正が行われている。」、署名者アクセス構造を  $A_s^* =$  「“経理部”AND“平社員”」として、自身が所持する「“経理部”、“平社員”」という属性集合に対応した秘密鍵  $SK_s$  を用いて、署名作成アルゴリズム  $SS, Sig$  を用いて署名  $\sigma^*$  を作成する。その後、アリスは  $m^*, \sigma^*, A_s^*$  を、社内の内部告発対応部署の署名検証者ボブへ送る。ボブは、署名検証を行う際に、署名者アクセス構造を  $A_s^* =$  「“経理部”AND“平社員”」とすると検証に通る(つまり、 $SS.Ver(PK_{ss}, \sigma^*, m^*, A_s^*) = 1$ ) が、一方で署名者アクセス構造を  $A'_s =$  「“人事部”AND“平社員”」としても検証に通ってしまう(つまり、 $SS.Ver(PK_{ss}, \sigma^*, m^*, A'_s) = 1$ ) 事実を発見する。そこで、ボブは、その平文  $m^*$  に対応する署名  $\sigma^*$  を本当に  $A_s^* =$  「“経理部”AND“平社員”」を満たす人物が作成したのかは“疑わしい”と判断し、その内部告発を正式に受理しない可能性がある。

従って、もしSP-ABS方式が署名者アクセス構造衝突困難性を満たさない場合、上で説明したような実用上の問題が生じる可能性があるため、既存のSP-ABS方式が当該性質を満たす可能性は高いという直感を抱いている。現時点ではまだ既存のSP-ABS方式が当該性質を満たすことの証明はどのSP-ABS方式に関しても完成していない。既存のSP-ABS方式が当該性質を満たすことの証明は、今後の課題として考えている。

## Chapter 9 結論

本研究では、暗号文ポリシー型属性ベース暗号(CP-ABE)と、署名者ポリシー型属性ベース署名(SP-ABS)の両機能を実現可能な、暗号文ポリシー型属性ベース Signcryption(CP-ABSC)に着目し、暗号文ポリシー型属性ベース鍵カプセル化メカニズム(CP-ABKEM), SP-ABS, データカプセル化メカニズム(DEM)を構成要素とするCP-ABSCの一般的構成法を提案した。そして、CP-ABKEMがAP-INC-CCA安全かつ復号者アクセス構造開示的であり、かつSP-ABSがAP-sEUF-CMA安全かつ完全匿名かつ署名者アクセス構造衝突困難であり、かつDEMがIND-CCA安全かつ一対一対応であるならば、CP-ABSCの安全性の中で最も強いと言われている、適応的述語モデルでのIND-CCA安全性(AP-IND-CCA), 適応的述語モデルでのsEUF-CMA安全性(AP-sEUF-CMA), 完全匿名性の三つの安全性を、提案するCP-ABSCの一般的構成法が達成可能であることを証明した。

また、それと関連した別の成果として、CP-ABEを構成要素としたCP-ABKEMの一般的構成法を提案した。そして、CP-ABEがAP-IND-CCA安全かつ復号者アクセス構造開示的であれば、CP-ABKEMの一般的構成はAP-IND-CCA安全かつ復号者アクセス構造開示的であることを証明した。

そして、これらの一般的構成法から既存のCP-ABSC方式よりも優位な性質を持ったCP-ABSC方式を具体的に構成できることを示した。具体的には、最強の安全性を達成可能であり、かつアクセス構造が非単調であり、かつ仮定が比較的弱い、という性質を持った最初のCP-ABSC方式を具体的に構成できることを示した。また、最強の安全性を達成可能であり、かつ“unbounded”であり、かつ仮定が比較的弱い、という性質をもった最初のCP-ABSC方式を具体的に構成できることを示した。また、本研究の成果は一般的構成であるので、今後効率や仮定の弱さ等に関して非常に優れた性質を持ったCP-ABE, SP-ABS等の新たな方式が提案された場合に、それらを本研究の一般的構成法の構成要素として利用することで、非常に優れた性質を持ったCP-ABSC方式が多数具体的に構成できるようになる可能性があるという利点があることも説明した。

また、本研究では、鍵ポリシー型属性ベース暗号(KP-ABE)と、鍵ポリシー型属性ベース署名(KP-ABS)の両機能を実現可能な、鍵ポリシー型属性ベース Signcryption(KP-ABSC)の一般的構成の提案も行い、KP-ABSCの安全性の中で最も強いと言われている、適応的属性モデルでのIND-CCA安全性(AA-IND-CCA), 適応的属性モデルでのsEUF-CMA安全性(AA-sEUF-CMA), 完全匿名性を達成可能であることを証明した。さらに、KP-ABEを構成要素としたKP-ABKEMの一般的構成法を提案し、KP-ABEがAA-IND-CCA安全かつ復号者属性集合開示的であれば、KP-ABKEMの一般的構成はAA-IND-CCA安全かつ復号者属性集合開示的であることを証明した。

## 謝辞

修士論文の作成にあたり、入学から現在に至るまで、ご指導下さりました東京大学生産技術研究所の松浦幹太教授に心から感謝致します。松浦先生の下では、研究に取り組む姿勢等基本的な事から、研究に関する幅広い専門知識など、数え切れないほど多くの有意義な物事の数々を学ぶことができました。あらゆる面で飛躍的に成長できた修士課程であったと確信しています。改めて深く感謝致します。

また、主に研究室打合せにおいて、研究内容に関する多くの有益な助言・質問をして下さったり、興味深い研究のアイデアをプレゼンテーションを通してシェアして下さった、今井秀樹先生、山口利恵先生、Mihaljevic Miodrag 先生、北川隆さん、北條孝佳さん、田村研輔さん、小林良輔さん、鈴木宏哉さん、疋田敏朗さん、宮野祐輔さん、崔誠云さんに、深く感謝致します。

そして、研究室のメンバーが生産技術研究所での研究活動を円滑に進める上で常日頃ご尽力下さった、現在研究室の秘書をされている、仲野小絵さん、佐伯麻紀さん、鶴山陽子さん、かつて秘書をされていた、小倉華代子さんに心より感謝致します。

さらに、松浦研究室の現在のメンバーである、細井琢郎さん、大畑幸矢さん、中田謙二郎さん、篠田詩織さん、竹之内玲さん、林昌吾さん、孫達さん、Tobias Fuchs さん、かつてメンバーであった、村上隆夫さん、Bongkot Jenjarrussakul さん、横手健一さん、馮菲さんには、主に研究室打合せにおいて、研究内容に関する生産的な議論をさせて頂きました。皆様のおかげで研究に対するモチベーションを常に高く保つことができました。心から感謝致します。特に大畑さんには、入学当初から現在に至るまで、研究内容に関して基礎的な知識から高度な専門知識まで幅広くご指導して頂きました。改めて深く感謝致します。

最後に、常日頃あらゆる面で大きな支えであった、家族、特に両親に心から感謝します。

## 参考文献

- [1] Gagné, M., Narayan,S., Safavi-Naini,R. :Threshold attribute-based signcryption. In:SCN 2010, LNCS 6280, pp. 154-171, 2010.
- [2] Emura,K., Miyaji,A., Rahman,M.S. :Dynamic attribute-based signcryption without random oracles. In:International Journal of Applied Cryptography, Vol. 2, No. 3, pp. 199-211, 2012.
- [3] Chen,C., Chen, J., Lim,H.W., Zhang, Z., Feng, D. :Combined public-key schemes: the cased of ABE and ABS. In:ProvSec 2012, LNCS 7496, pp.53-69, 2012.
- [4] Pandit, T., Pandey, S.K., Barua, R. :Attribute-based signcryption: signer privacy, strong unforgeability, and IND-CCA2 security in adaptive-predicates attack. In:ProvSec 2014, LNCS 8782, pp.274-290, 2014.
- [5] Pandit, T., Pandey, S.K., Barua, R. :Attribute-based signcryption :signer privacy, strong unforgeability and IND-CCA2 security in adaptive-predicates attack. Cryptology ePrint Archive 2015/555, 2015.
- [6] Rao, Y.S., Dutta, R. :Expressive bandwidth-efficient attribute based signature and signcryption in standard model. In:ACISP 2014, LNCS 8544, pp. 209-225, 2014.
- [7] Datta, P., Dutta, R., Mukhopadhyay, S. :Functional signcryption: notion, construction, and applications. In:ProvSec 2015, LNCS 9451, pp.268-288, 2015.
- [8] Datta, P., Dutta, R., Mukhopadhyay, S. :Functional signcryption: notion, construction, and applications. Cryptology ePrint Archive 2015/913, 2015.
- [9] Nandi, M., Pandit, T. :On the power of pair encodings: frameworks for predicate cryptographic primitives. Cryptology ePrint Archive 2015/955, 2015.
- [10] Attrapadung, N. :Dual system encryption via doubly selective security: framework, fully-secure functional encryption for regular languages, and more. In:EUROCRYPT 2014, LNCS 8441, pp. 557-577, 2014.
- [11] Attrapadung, N. :Dual system encryption via doubly selective security: framework, fully-secure functional encryption for regular languages, and more. Cryptology ePrint Archive 2014/428, 2014.
- [12] Emura, K., Miyaji, A., Nomura, A., Omote, K., Soshi, M.:A ciphertext-policy attribute-based encryption scheme with constant ciphertext length. In:ISPEC 2009, LNCS 5451, pp. 13-23, 2009.

- [13] Chen, C., Zhang, Z., Feng, D.:Efficient ciphertext policy attribute-based encryption with constant-size ciphertext and constant computation-cost. In:ProvSec 2011, LNCS 6980, pp. 84-101, 2011.
- [14] Pandit, T, Barua, R.:Efficient fully secure attribute-based encryption schemes for general access structures. In:ProvSec 2012, LNCS 7496, pp. 193-214, 2012.
- [15] Zhang, Y, Zheng, D., Chen, X., Li, J., Li, H.:Computationally efficient ciphertext-policy attribute-based encryption with constant size ciphertexts. In:ProvSec 2014, LNCS 8782, pp. 259-273, 2014.
- [16] Rouselakis, Y., Waters, B. :Practical constructions and new proof methods for large universe attribute-based encryption. In: ACM CCS, pp. 463-474, 2013.
- [17] Lewko, A.B., Waters, B. :Decentralizing attribute-based encryption. In:EUROCRYPT 2011, LNCS 6632, pp. 568-588, 2011.
- [18] Yamada, S., Attrapadung, N., Hanaoka, G., Kunihiro, N. :A framework and compact constructions for non-monotonic attribute-based encryption. In:PKC 2014, LNCS 8383, pp. 275-292, 2014.
- [19] Yamada, S., Attrapadung, N., Hanaoka, G., Kunihiro, N. :A framework and compact constructions for non-monotonic attribute-based encryption. Cryptology ePrint Archive 2014/181, 2014.
- [20] Okamoto, T., Takashima, K. :Fully secure functional encryption with general relations from the decisional linear assumption. In:CRYPTO 2010, LNCS 6223, pp. 191-208, 2010.
- [21] Okamoto, T., Takashima, K. :Fully secure functional encryption with general relations from the decisional linear assumption. Cryptology ePrint Archive 2010/563, 2010.
- [22] Okamoto, T., Takashima, K. :Fully secure unbounded inner-product and attribute-based encryption. In:ASIACRYPT 2012, LNCS 7658, pp. 349-366, 2012.
- [23] Okamoto, T., Takashima, K. :Fully secure unbounded inner-product and attribute-based encryption. Cryptology ePrint Archive 2012/671, 2012.
- [24] Attrapadung, N., Yamada, S.:Duality in ABE:Converting attribute based encryption for dual predicate and dual policy via computational encodings. In:CT-RSA 2015, LNCS 9048, pp. 87-105, 2015.
- [25] Attrapadung, N., Yamada, S.:Duality in ABE:Converting attribute based encryption for dual predicate and dual policy via computational encodings. Cryptology ePrint Archive 2015/157, 2015.

- [26] Maji, H.K., Prabhakaran, M., Rosulek, M. :Attribute-based signatures: achieving attribute-privacy and collusion-resistance. Cryptology ePrint Archive 2008/328, 2008.
- [27] Maji, H.K., Prabhakaran, M., Rosulek, M. :Attribute-based signatures. In:CT-RSA 2011, LNCS 6558, pp. 376-392, 2011.
- [28] Okamoto, T., Takashima, K. :Efficient attribute-based signatures for non-monotone predicates in the standard model. In:PKC 2011, LNCS 6571, pp. 35-52, 2011.
- [29] Okamoto, T., Takashima, K. :Efficient attribute-based signatures for non-monotone predicates in the standard model. Cryptology ePrint Archive 2011/700, 2011.
- [30] Matsuda, T., Matsuura, K., Schuldt, J.C.N. :Efficient constructions of signcryption schemes and signcryption composability. In:INDOCRYPT 2009, LNCS 5922, pp. 321-342, 2009.
- [31] Chiba, D., Matsuda, T., Schuldt, J.C.N, Matsuura, K. :Efficient generic construction of signcryption with insider security in the multi-user setting. In:ACNS. LNCS 6715, pp. 220-237, 2011.
- [32] Zhao, F., Nishide, T., Sakurai, K. :Realizing fine-grained and flexible access control to outsourced data with attribute-based cryptosystems. In:ISPEC 2011. LNCS 6672, pp. 83-97, 2011.
- [33] Wang, C., Xu, X., Li, Y., Shi, D. :Integrating ciphertext-policy attribute-based encryption with identity-based ring signature to enhance security and privacy in wireless body area networks. In:Inscrypt 2014. LNCS 8957, pp. 424-442, 2015.
- [34] Anada, H., Arita, S., Handa, S., Iwabuchi, Y. :Attribute-based identification: definitions and efficient constructions. In:ACISP 2013, LNCS 7959, pp.168-186, 2013.
- [35] Blämer, J., Liske, G. :Direct chosen-ciphertext secure attribute-based key encapsulations without random oracles. Cryptology ePrint Archive 2013/646, 2013.
- [36] Gorantla, M. C., Boyd, C., Nieto, J. M. G. :Attribute-based authenticated key exchange. In: ACISP 2010, LNCS 6168, pp. 300-317, 2010.
- [37] Gorantla, M. C., Boyd, C., Nieto, J. M. G. :Attribute-based authenticated key exchange. Cryptology ePrint Archive 2010/084, 2010.
- [38] Wang, C.-J., Huang, J.-S., Lin, W.-L., Lin, H.-T. :Security Analysis of Gagné et al.'s threshold attribute-based signcryption scheme. In:INCoS 2013, pp. 103-108, 2013.
- [39] Zheng, Y. :Digital signcryption of how to achieve  $\text{cost}(\text{signature} \ \& \ \text{encryption}) \ll \text{cost}(\text{signature})+\text{cost}(\text{encryption})$ . In:CRYPTO 1997, LNCS 1294, pp. 165-179, 1997.

- [40] Yamada, S., Attrapadung, N., Hanaoka, G., Kunihiro, N. :Generic constructions for chosen ciphertext secure attribute based encryption. In:PKC 2011. LNCS 6571, pp. 71-89, 2011.
- [41] Sahai, A., Waters, B.:Fuzzy identity-based encryption. In:EUROCRYPT 2005, LNCS 3494, pp. 457-473, 2005.
- [42] Goyal, V., Pandey, O., Sahai, A., Waters, B. Attribute-based encryption for fine-grained access control of encrypted data. In:ACM CCS 2006, pp. 89-98, 2006.
- [43] Bethencourt, J., Sahai, A., Waters,B. :Ciphertext-policy attribute-based encryption. In:IEEE Symposium on Security and Privacy 2007, pp. 321-334, 2007.
- [44] Shoup, V. :Using hash functions as a hedge against chosen ciphertext attack. In:EUROCRYPT 2000, LNCS 1807, pp. 275-288, 2000.
- [45] Diffie, H., Hellman, M. :New Directions in cryptography. In:Information Theory, vol.22, pp.644-654, 1976.

# 発表文献

## 国内会議

- i 石坂理人, 大畑幸矢, 松浦幹太. :適応的述語安全な暗号文ポリシー型属性ベース Signcryption の一般的構成. In:2016年 暗号と情報セキュリティシンポジウム (SCIS2016) 予稿集, 2C3-4, 熊本, 1月, 2016年.