修士論文

多人数モデルで内部者安全なSigncryption に関する研究

A Study on Signcryption with Insider Security in the Multi-user Setting

指導教員　　松浦幹太　准教授

東京大学大学院 情報理工学系研究科 電子情報学専攻

48-096414　　千葉 大輝

平成23年2月9日提出

内容梗概

近年のインターネットの急速な普及により，情報資源が大きな価値を持つようになり, 世界中の情報がネットワークを介して伝送されるようになってきている．それに伴って情報セキュリティの重要性が高まり, 情報の秘匿性, 及び情報の完全性を確保するための手段として暗号技術が用いられ, 安全な暗号技術は情報化社会において必要不可欠な存在となっている.

この情報の秘匿性と完全性を同時に満たす暗号技術として, Signcryption がある. Signcryption は公開鍵暗号と電子署名を機能を併せ持った暗号技術で, 秘匿性と完全性を同時に保守しながら通信を行いたい場合に, 単純に電子署名と公開鍵暗号を組み合わせるよりも遥かに効率が良い, という利点がある. Signcryption が利用された場合に想定される攻撃者として, “外部者”, “内部者” という分類がある. さらに, 公開鍵暗号や署名の場合とは異なり, 一人の送信者と一人の受信者からなる “二人モデル” での安全性は, 多人数でのモデルの安全性を包含しないため, 多人数モデルでの安全性を考えるのが望ましい. しかし, 我々の知る限り, 多人数を考慮した内部者に対する秘匿性と “強” 偽造不可能性をスタンダードモデルで達成する手法がこれまでに存在しなかった. そこで我々はそれらを達成する最初の手法として, Signcryption の一般的構成法を二つ提案する. 一つ目の手法ではタグベース KEM を用い, 二つ目の手法では KEM を用いる. これらの手法により, 多人数モデルで内部者安全な Signcryption を具体的に数多く構成できる.

また, 本稿ではもう一つの研究成果として Signcryption の一般的構成法の要素技術であるタグベース KEM の強化手法を提案する．タグベース KEM(TBKEM) とは, 暗号化と復号アルゴリズムに補助入力として任意のタグを入力できる KEM である. TBKEM の CCA 安全性は, 攻撃者がタグを選ぶタイミングにより, 選択的タグ版と適応的タグ版の二種類が考えられる．本稿では, カメレオンハッシュを用いた選択的タグ CCA 安全な TBKEM を適応的タグ CCA 安全な TBKEM への強化手法を提案する．既に同目的を達成する自明な手法が存在するが, 本手法は自明な手法と比べ変換後の暗号文サイズを小さくできる．本手法を Signcryption の一般的構成法に適用させることで, 選択的タグ安全な TBKEM だけでなく適応的タグ安全な TBKEM も Signcryption の要素技術として利用できるため, Signcryption の具体的な構成にも非常に有用である.

目次

Chapter 1　序論

1.1 Signcryption

近年のインターネットの急速な普及により，情報資源が大きな価値を持つようになり, 世界中の情報がネットワークを介して伝送されるようになってきている．それに伴って情報セキュリティの重要性が高まっている．インターネットというオープンなネットワーク上を流れる情報は, 悪意のある攻撃者による盗聴や改竄などの危険性がつきまとっている. このような攻撃が可能な環境においても, 情報の秘匿性, 及び情報の完全性を確保するための手段として暗号技術が用いられており, 安全な暗号技術は情報化社会において必要不可欠な存在となっている.

情報を暗号化して送受信する際, 送信者と受信者が互いに同じ鍵を用いる共通鍵暗号方式を利用する場合, 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の問題が生じる．この問題を解決するのが公開鍵暗号である. 公開鍵暗号方式 (Public Key Encryption, PKE) [21] とは，暗号化する際に用いる公開鍵と，復号する際に用いる秘密鍵が異なるため，暗号化鍵を公開することができる．すなわち, 事前に鍵を共有せずとも，容易に鍵配送を行うことができる.

ハイブリッド暗号とタグベース KEM　公開鍵暗号は, 1976 年の Diffie と Hellman による安全な鍵共有の概念の提唱 [21] に始まる. この発明以降, その構成法や安全性の議論が活発に行われてきた. その一つに, 平文を暗号化するために使用する共通鍵そのものを公開鍵暗号方式を使用して暗号化する, ハイブリッド暗号と呼ばれる暗号方式がある. ハイブリッド暗号は Shoup [39] によって 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 (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/復号する方式を, データカプセル化メカニズム (DEM) は共通鍵暗号方式で平文を暗号化/復号する方式を表す. これにより, 公開鍵暗号の利点である鍵配送の問題の解決, ならびに共通鍵暗号の利点である計算速度の速さの双方を共に実現可能にしている.

また, 公開鍵暗号方式の安全として十分といわれている選択暗号文攻撃に対して安全 (CCA 安全, 詳細は 2.2 節で述べる) な公開鍵暗号 (PKE) や鍵カプセル化メカニズム (KEM) を他の要素技術を構成するための構成要素として使用する場合, “タグ” つきの PKE や KEM を用いれば便利な場合が多い. [1] タグつきの PKE(タグベース暗号, TBE) [35, 29] や KEM(タグベース KEM, TBKEM) [2] は, 暗号化, 復号アルゴリズ

[1] “ラベル” 付き暗号という名で呼ばれることもある [40].

[2] 本稿で考えている TBKEM は, Abe ら [4] で定義された Tag-KEM とは異なる要素技術である. 詳細は 2.2 節の定義を参照されたい.

ムにおいて追加の補助入力として任意の文字列 “タグ” をとる．そして, 暗号化と復号の時に同一のタグを用いたときに正しく復号されることが保証される．タグベースの要素技術の応用先の例は, メッセージの多重暗号化 [22], タイムリリース暗号 [19], Signcryption [45] の構成 [36] などがある．

TBE や TBKEM の CCA 安全性を考える場合, 攻撃者がタグを選ぶタイミングにより, 選択的タグ版と適応的タグ版の 2 種類が考えられる．安全性として強いのは後者の安全性を持つ方であり, それだけ応用先が広い．従って, 選択的タグ CCA 安全な TBE や TBKEM から, 適応的タグ CCA 安全な TBE や TBKEM を少ないコストで構成することができれば, 非常に有用である．

電子署名　また, 1976 年に Diffie と Hellman によって, メッセージの完全性を保証する暗号技術として, 公開鍵暗号と共に電子署名の概念が提唱された [21]．電子署名方式では, 署名者は自身の秘密鍵を用いてメッセージに署名を発行し, 検証者は署名者の公開鍵を用いて署名が正しいものであるかどうかの検証を行う．秘密鍵を知らない第三者は, 他の署名とそれに対応するデータを複数参考にしても, 正しい署名を作る事ができない．また, 同じ秘密鍵によって署名すると, 同じ署名が生成されるような異なる複数のデータを見つけることは困難である, という事が要求される．これらの安全性は “偽造不可能性” として定式化されている (詳細は 2.4.1 節において述べる)．このような機能を持つ電子署名の応用先として, 公開鍵基盤 (PKI) において各公開鍵と受信者が正しく対応している事を認証する役割などがあげられる．

Signcryption　1997 年には Zheng によって, 公開鍵暗号と電子署名を組み合わせた機能を持つ, Signcryption と呼ばれる暗号技術が提案された [45]．Signcryption が保守する安全性は, 平文の秘匿性, メッセージの偽造不可能性, さらに否認防止である．[45] の主な成果は, これらの安全性を満たした通信を行う場合に, 単純に電子署名と公開鍵暗号を組み合わせるよりも Signcryption を利用した方が遥かに効率が良い, という点にあった．

Signcryption 方式では, 送信者は受信者の公開鍵を用いて送信したいメッセージを暗号化し, 同時に自身の秘密鍵を用いてメッセージに署名を発行する．受信者は送られてきた情報を自身の秘密鍵を用いて復号し, 同時に送信者の公開鍵を用いて署名の検証を行う．また, Signcryption のアルゴリズムは, 暗号化と署名を行う順序から, Sign-then-Encrypt と Encrypt-then-Sign の二つに分けることができる (本稿の提案手法は Sign-then-Encrypt である)．Signcryption の概観を図 1.1 に示す．

Signcryption の安全性モデルを議論する上で, 最もシンプルなモデルは一人の送信者と一人の受信者からなるいわゆる二人モデルである．しかし, 通常の公開鍵暗号や電子署名とは異なり, Signcryption では二人モデルでの安全性は多人数モデルでの安全性を包含しない．従って, Signcryption の安全性モデルとして多人数モデルを考える事は非常に重要である．本稿でも多人数環境での安全性のみ議論している．

また, もう一つの Signcryption の安全性定義の重要な要素として, 攻撃者が送信者や受信者を攻撃する外部者なのか, 送信者または受信者の役割の一部を担う内部者な

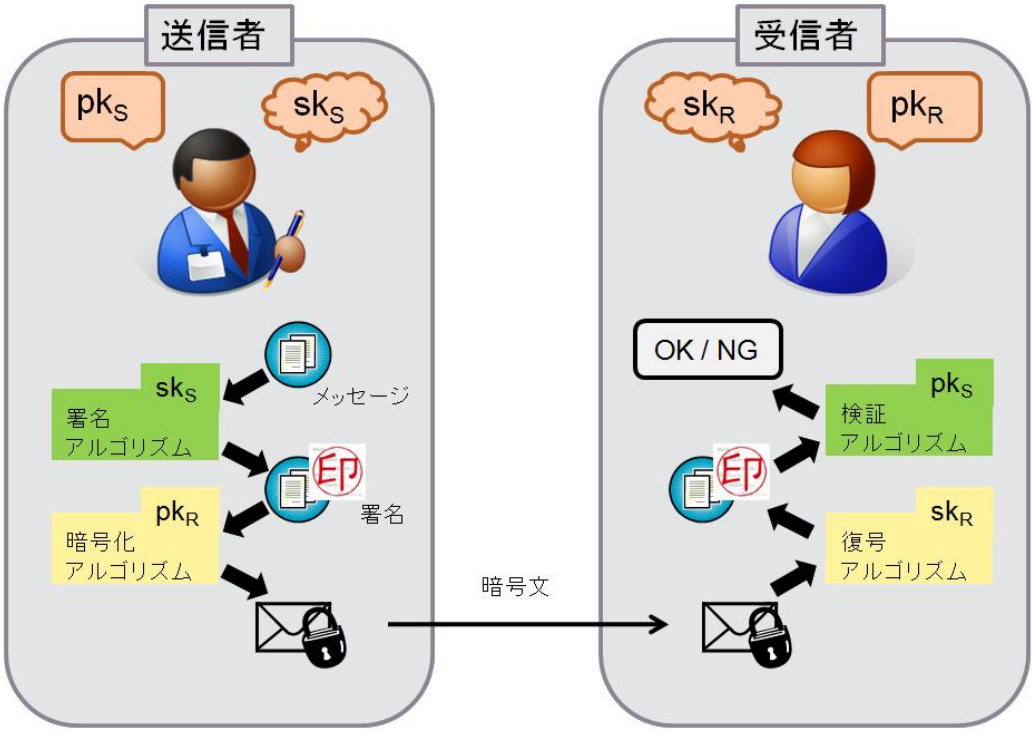

図 1.1: Signcryption(Sign-then-Encrpt 方式)

のか, という点がある. 攻撃者が内部者である事の定義は, 攻撃者の目標が秘匿性を攻撃するのか偽造不可能性を攻撃するかによって異なる. 内部者で秘匿性を攻撃する攻撃者を考える場合, 攻撃者が送信者/受信者の公開鍵 (pk_S, pk_R) だけでなく送信者の秘密鍵 (sk_S) も知っている状況を想定する (図 1.2). 内部者で偽造不可能性を攻撃する攻撃者を考える場合, 攻撃者が送信者/受信者の公開鍵 (pk_S, pk_R) だけでなく受信者の秘密鍵 (sk_R) も知っている状況を想定する (図 1.3).

外部者安全性より内部者安全性の方が強いため, Signcryption が内部者安全性を達成する方が望ましい.

1.2　証明可能安全性

あらゆる暗号技術は, 現在知られている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を直接意味するわけではない. しかし, 客観的な安全性の議論を行うために, 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.

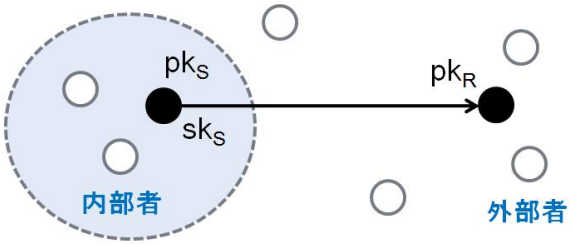

図 1.2: 秘匿性を攻撃する攻撃者

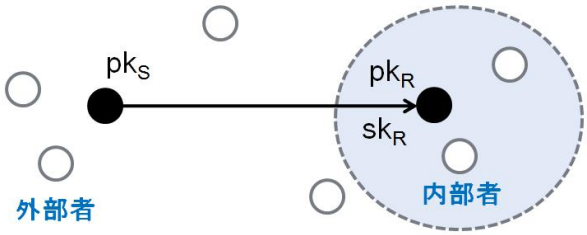

図 1.3: 偽造不可能性を攻撃する攻撃者

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の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 “現実に知られている難しい問題の困難性の仮定が成り立つならば, 安全性を無視できない確率で破るアルゴリズムが存在しない”, というものになっている. 証明の際には, その対偶を示すことで行われる. 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, 現実に知られている困難な問題を解くことができる” という事を示すことにより, 問題の困難性の仮定を破ることから, 背理法により安全性を破る攻撃者は存在しない, という手法を利用する. 確率多項式時間アルゴリズムは, 現実に存在するアルゴリズムの能力を表しているおり, 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 章および 3 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場合が多い). 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.

ランダムオラクルモデルとスタンダードモデル　暗号方式の安全性の 証明において, ランダムオラクルモデル (Random Oracle Model) [23, 10] と呼ばれるモデルが使用されることがある. ランダムオラクルとは, 入力であるあらゆる問合せに値域に真にランダムに分布するような応答を返すが, 同じ問合せに対しては毎回同じ応答をするオラクル (理論的ブラックボックス) である. これは, 出力空間において一様分布であるとみなせるハッシュ関数であるとも考えることができる.

ハッシュ関数の安全性の一つに衝突困難性があるが, 実用的なハッシュ関数では衝突困難性を満たすことは証明されていない. そこで, Bellare と Rogaway は, ランダムオラクルモデルというモデルをはじめて定式化した [10]. ランダムオラクルモデルとは, ランダムオラクルが存在すると仮定するモデルのことである. これに対し, ランダムオラクルを使用しないモデルをスタンダードモデル (Standard Model) という. 実用の際には, ランダムオラクルの部分には SHA-1 や SHA-256 などの実用的なハッシュ関数を使用する. 一般的に, ランダムオラクルを使う方式の方が, スタンダードモデルでの方式よりも計算コストや署名サイズなどの面で効率のよいものができる. 実際に使用されている DSA [1], RSA [2], あるいは RSA-PSS [11] などといった多くの方式がランダムオラクルモデルでのものである.

ランダムオラクルはあくまで理想的な存在であり, 現実世界においてはランダムオラクルのような真にランダムな出力を持つ関数は存在しない. また, ランダムオラクルモデルにおいて証明可能安全性を有する方式は, そのランダムオラクル以外のどのような関数に置き換えても, スタンダードモデルでは安全性を証明できなくなるものも多く存在することが分かっている [18, 37].

このような理由により, 新しく暗号学的な方式を考える際は, ランダムオラクルを使わなくてもよいならば, ランダムオラクルを用いないスタンダードモデルでの方式を目指す事が望ましい. しかし暗号文サイズや計算コストの面でスタンダードモデルはランダムオラクルモデルに劣るため, 実用化を考えるとどうしても速さなどが欲しい場合にはランダムオラクルモデルでの方式を，絶対に守らなければならない用途に使用する場合にはスタンダードモデルでの方式を使用する，といった使い分けが必要になると考えられる.

1.3　一般的構成法

一般的構成法とは, ある暗号方式を, その構成要素として他の暗号技術を組み合わせるなどして構成する手法である. 一般的構成法においては, 1.2 節で述べたような現実に知られている具体的な数論の困難性 (素因数分解問題や離散対数問題など) ではなく, 構成要素となる暗号方式をブラックボックスとして用いて, それぞれの安全性要件を構成する暗号方式の安全性の根拠として用いる. つまり, 既に安全性要件を満たしてい

る構成要素となる暗号方式があれば, それらを組み合わせることで新たな暗号方式を構成することが可能になる.

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

それだけでなく, 一般的構成法によって構成された暗号方式は, 暗号の危殆化に対して耐性があるという利点もある. 一般的構成法においては, 構成要素となる暗号方式がそれぞれの安全性要件を満たしていさえすればどのような暗号方式でも利用可能であるため, 数多くの具体的な構成が可能となる. これにより, 例えば素因数分解を解く効率的なアルゴリズムが発見されてしまった場合でも, 離散対数系の問題を安全性の拠り所としている要素技術を代替として利用することで, 安全性の確保が可能となる.

以上より, 暗号技術の構成法として一般的構成法を用いることは非常に有用であると言える.

1.4　本研究の貢献

本研究の貢献は, 大きく分けてタグベース KEM の安全性の強化手法の提案と, 多人数モデルにおける内部者安全な Signcryption の一般的構成法の提案の 2 点である.

TBKEM の強化手法　カメレオンハッシュと呼ばれる特殊なハッシュ関数を用いて, 選択的タグ CCA 安全な TBKEM から適応的タグ CCA 安全な TBKEM への効率的な強化手法を提案する. カメレオンハッシュを用いる動機は 2.6 節で述べる. 選択的タグ CCA 安全な TBKEM から適応的タグ CCA 安全な TBKEM への変換については, Kiltz [29] の PKE から TBE への変換などを用いることで同目的を達成する自明な手法が存在するが, 本手法は自明な手法と比べ変換後の暗号文サイズを小さくできるため, 変換後の暗号文サイズの増加を抑えたい場合には本提案手法は便利である. 詳細は 5 章を参照されたい.

また, この提案手法の応用先として, スタンダードモデルで内部者に対する安全性を証明可能な Signcryption の構成を考える. 最近 [36] において TBKEM のうち特殊な性質を持つものを構成要素の一つとする Signcryption の構成法が提案された. 彼らの構成法では, TBKEM が適応的タグ CCA 安全ならば最も強い秘匿性を, 選択的タグ CCA 安全であればそれよりも弱いモデルでの安全性を達成できることが示されている. これに対し本強化手法では彼らの Signcryption 方式で TBKEM に必要とされる “分割可能性” という性質と, ある電子署名との TBKEM との間で考えられる性質である “Signcryption 結合可能性” と呼ばれる性質が保存され, 変換後の TBKEM とともに [36] の Signcryption 結合可能性を持つ署名方式が存在する. 従って, 既に [36] で示された多くの選択的タグ CCA 安全な TBKEM に我々の変換を適用すれば, 今までより

もより多くの強いモデルでの安全性をスタンダードモデルで証明可能な Signcryption 方式が構成できる.

さらに, 応用先として考えられる Signcryption の構成は [36] の手法だけではなく, IND-tag-CCA 安全な TBKEM を構成要素とする一般的構成法であればどのような Signcryption の構成法にも適用可能である. 後述する本稿におけるもう一つの研究成果である Signcryption の一般的構成法は構成要素として IND-tag-CCA 安全な TBKEM を要求するため, この強化手法が利用可能である.

多人数モデルにおける内部者安全な Signcryption の一般的構成法　我々の知る限り, 多人数モデルで (特殊な仮定なしに) 内部者に対する秘匿性と強偽造不可能性をスタンダードモデルで達成する Signcryption の構成法は存在しない. そこで本稿では, そのような Signcryption の一般的構成法を “二つ” 提案する. この構成法は多人数環境で内部者に対する “強” 偽造不可能性と識別不可能性をあらゆる制限なしにスタンダードモデルで達成する初めての方式である.

第一の方式 (SC_{tk}) は IND-tag-CCA 安全な TBKEM, 1 対 1 対応の性質を持つ IND-CCA 安全な DEM, 及び sUF-CMA 安全な署名を用いた (詳細な定義は 2 章を参照). 提案した構成法は Sign-then-Encrypt の一種と考える事ができる. 過去の研究によって Sign-then-Encrypt 方式では強偽造不可能性が達成できないことが分かっている [6, 36]. しかし提案手法では, KEM/DEM フレームワークを採用する事によって単に平文のみに対して署名するのでなく, 暗号文の一部 (KEM の暗号文) に対しても署名する. それによって攻撃者が強偽造不可能性を破るには DEM 部を改変して辻褄の合う偽造された暗号文を作るしかないが, DEM が 1 対 1 対応を満たせばこれも防ぐ事が出来る. 幸いな事に 1 対 1 を満たす DEM を構成する事は容易である.

第二の方式 (SC_{kem}) は IND-CCA 安全な KEM, 1 対 1 対応の性質を持つ IND-OT 安全な DEM, 1 対 1 対応の性質をもつ一回安全な MAC, 及び sUF-CMA 安全な署名を用いる. 基本的なアイデアは第一の方式とほぼ同様である. 要素技術として TBKEM を使わない代わりに, KEM と MAC を用いる. IND-tag-CCA 安全な TBKEM は IND-CCA 安全な KEM と sUF-OT 安全な MAC から構成可能で [4], IND-CCA 安全な DEM は IND-OT 安全な DEM と sUF-OT 安全な MAC から構成可能である [9] と分かっている. SC_{kem} はこのように MAC を使って構成された TBKEM と DEM を SC_{tk} のそれらにあてはめて, MAC の部分を共有した形になっている.

以上 2 つの構成法から, スタンダードモデルで多人数環境における内部者安全な Signcryption を数多く構成できる. それを示すため, 7 章において既存のスタンダードモデルの構成法とともに提案手法を用いた具体的な Signcryption の比較を行う.

[36] と同様に, 上記手法の利点は提案手法から効率的な Signcryption の構成ができるだけでなく, “一般的構成法”, すなわち既に確立された構成要素の安全性の結果を利用できるという点にもあると強調したい.

1.5　本稿の構成

以下, 2 章では, 3 章以降の提案方式の説明のときに必要になる暗号方式のアルゴリズムやその安全性などの諸定義を概要と共に説明する. 3 章では, 本研究で構成する Signcryption の方式と, その安全性定義を示す. 4 章では, 本研究の関連研究を紹介する. 5 章では, TBKEM の強化手法の提案方式と, 既存の強化手法との比較を行う. 6 章では, Signcryption の一般的構成法のいくつかの提案方式と, それらの安全性証明を行う. 7 章では, 5 章での提案方式に関する進んだ議論と, 6 章での提案方式に関する効率の比較を行う. 8 章は本稿のまとめである.

なお, 5 章の提案方式を中心とする本稿の内容の一部は, 査読無し国内会議 SCIS 2010(発表文献 [ii]) において発表した. また, 6 章の提案方式を中心とする本稿の内容の一部は, 査読無し国内会議 SCIS 2011(発表文献 [iii]) において発表した.

Chapter 2　諸定義

本稿で使用する言葉や安全性の定義を行う. まず, $x \leftarrow y$ と書くとき, y が集合ならばそこから一様ランダムに要素を取り出し x に代入する操作を, y がアルゴリズムまたは関数ならば x を出力する操作を表す. "$x||y$" は x と y の連結を表し, $\mathcal{K}$, $\mathcal{R}$ はそれぞれ鍵空間, 乱数空間を表す. また, $\mathcal{A}$ が確率的アルゴリズムであり, $y \leftarrow \mathcal{A}(x;r)$ と書くとき, $\mathcal{A}$ は x を入力, r を乱数として用いて計算し, y を出力することを意味する. $\Pr[x]$ とは, x が起こる確率を表すこととし, "κ" は常にセキュリティパラメータを表す. また, ある関数が無視できる値であるという場合, セキュリティパラメータ κ について無視できる値である事を意味する.

2.1　鍵カプセル化メカニズム (KEM)

鍵カプセル化メカニズム (Key Encapsulation Mechanism. 以下, KEM と表記する) とは, 公開鍵暗号と共通鍵暗号の長所を組み合わせた暗号方式であるハイブリッド暗号のうち, 乱数を含む公開鍵暗号の部分を定式化したものである.

KEM は以下の 4 つのアルゴリズムから成る.

KSetup: セキュリティパラメータ 1^κ を入力とし, 公開パラメータ prm を出力する.

KKG: prm を入力とし, 公開鍵/秘密鍵の対 (pk, sk) を出力する.

Encap: prm, pk を入力とし, 暗号文 c, 共有鍵 K を出力する.

Decap: prm, sk, c を入力とし, K(あるいは復号失敗 "$\perp$") を出力する.

全ての $prm \leftarrow \mathsf{KSetup}(1^\kappa)$, $(pk, sk) \leftarrow \mathsf{KKG}(prm)$, $(c, K) \leftarrow \mathsf{Encap}(prm, pk)$ に対し, $K = \mathsf{Decap}(prm, sk, c)$ が求められる.

2.1.1　IND-CCA 安全性

本稿では, KEM の安全性として最も強いとされている "適応的選択暗号文攻撃に対する識別不可能性"(IND-CCA) を取り扱う. KEM KM の IND-CCA 安全性は, 以下の攻撃者 $\mathcal{A}$ と IND-CCA チャレンジャー $\mathcal{CH}$ 間の IND-CCA ゲームを用いて定義される.

Setup. $\mathcal{CH}$ は $prm \leftarrow \mathsf{KSetup}(1^\kappa)$, $(pk, sk) \leftarrow \mathsf{KKG}(prm)$ を計算する. 出力された (prm, pk) を $\mathcal{A}$ に渡し, sk を保持しておく.

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

Challenge. $\mathcal{A}$ は暗号文の要求を $\mathcal{CH}$ に送る. $\mathcal{CH}$ は $(c^*, K_1^*) \leftarrow \mathsf{Encap}(prm, pk)$ を計算し, $K_0^* \in \mathcal{K}$ を一様ランダムに選ぶ. 次にランダムにコイン $b \in \{0, 1\}$ を振り, これら (c^*, K_b^*) を $\mathcal{A}$ に渡す.

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

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

ここで, ある KEM KM に対する $\mathcal{A}$ のアドバンテージを以下の様に定義する.

$$\mathsf{Adv}^{\text{IND-CCA}}_{KM,\,\mathcal{A}} = |\Pr[b' = b] - \frac{1}{2}|$$

定義 2.1. 全ての多項式時間アルゴリズム $\mathcal{A}$ に対し, $\mathsf{Adv}^{\text{IND-CCA}}_{KM,\,\mathcal{A}}$ が無視できる値であるとき, *KEM* KM は *IND-CCA* 安全であるという.

2.2　タグベース鍵カプセル化メカニズム (TBKEM)

タグベース鍵カプセル化メカニズム (タグベース KEM. 以下, TBKEM と表記する) とは, KEM の Encap と Decap アルゴリズムにおいて追加の補助入力として “タグ” と呼ばれる任意の文字列をとる暗号方式である. 暗号化と復号の時に同一のタグを用いたときに正しく復号されることが保証される. これはタグベース暗号 [35, 29] の KEM 版と考える事もできる. ただし, Abe らによって提案されたハイブリッド暗号の要素技術である Tag-KEM [4] とは異なることに注意されたい.

TBKEM は以下の 4 つのアルゴリズムから成る.

TSetup: セキュリティパラメータ 1^κ を入力とし, 公開パラメータ prm を出力する.

TKG: prm を入力とし, 公開鍵/秘密鍵の対 (pk, sk) を出力する.

TEncap: prm, pk, タグ tag, を入力とし, 暗号文 c, 共有鍵 K を出力する.

TDecap: prm, sk, tag, c を入力とし, K(あるいは復号失敗 “$\perp$”) を出力する.

全ての $prm \leftarrow \mathsf{TSetup}(1^\kappa)$, $(pk, sk) \leftarrow \mathsf{TKG}(prm)$, tag, $(c, K) \leftarrow \mathsf{TEncap}(prm, pk, \mathsf{tag})$ に対し, $K = \mathsf{TDecap}(prm, sk, \mathsf{tag}, c)$ が求められる.

2.2.1　IND-tag-CCA 安全性

本稿では, TBKEM の安全性として, 攻撃者がタグを選ぶタイミングにより, 適応的タグ版 (IND-tag-CCA 安全性) と選択的タグ版 (適応的タグ CCA 安全性, IND-stag-CCA 安全性) の 2 種類を取り扱う.

まず, TBKEM の安全性として最も強いとされている “適応的タグ及び適応的選択暗号文攻撃に対する識別不可能性”(IND-tag-CCA) について述べる. TBKEM TK

の IND-tag-CCA 安全性は, 以下の攻撃者 $\mathcal{A}$ と IND-tag-CCA チャレンジャー $\mathcal{CH}$ 間の IND-tag-CCA ゲームを用いて定義される.

Setup. $\mathcal{CH}$ は $prm \leftarrow \mathsf{TSetup}(1^\kappa)$, $(pk, sk) \leftarrow \mathsf{TKG}(prm)$ を計算する. 出力された (prm, pk) を $\mathcal{A}$ に渡し, sk を保持しておく.

Phase 1. $\mathcal{A}$ は $\mathcal{CH}$ に対し, 任意の回数, 復号クエリ (tag, c) を発行することができる. $\mathcal{CH}$ はそれぞれのクエリ (tag, c) に対し, 正しい復号結果 K / $\perp$ $\leftarrow \mathsf{TDecap}(prm, sk, \mathsf{tag}, c)$ を返す.

Challenge. $\mathcal{A}$ は任意のタグ tag^* を選び, $\mathcal{CH}$ に送る. $\mathcal{CH}$ は tag^* に対する暗号文 $(c^*, K_1^*) \leftarrow \mathsf{TEncap}(prm, pk, \mathsf{tag}^*)$ を計算し, $K_0^* \in \mathcal{K}$ をランダムに選ぶ. 次にランダムにコイン $b \in \{0, 1\}$ を振り, これら (c^*, K_b^*) を $\mathcal{A}$ に渡す.

Phase 2. $\mathcal{A}$ は Phase 1. と同様に復号クエリを発行することができる. ただし, $\mathcal{A}$ は (tag^*, c^*) を復号クエリとすることはできない.

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

ここで, ある TBKEM TK に対する $\mathcal{A}$ のアドバンテージを以下の様に定義する.

$$\mathsf{Adv}^{\text{IND-tag-CCA}}_{TK,\ \mathcal{A}} = |\Pr[b' = b] - \frac{1}{2}|$$

定義 2.2. 全ての多項式時間アルゴリズム $\mathcal{A}$ に対し, $\mathsf{Adv}^{\text{IND-tag-CCA}}_{TK,\ \mathcal{A}}$ が無視できる値であるとき, *TBKEM* TK は *IND-tag-CCA* 安全であるという.

2.2.2　IND-stag-CCA 安全性

TBKEM の安全性として IND-tag-CCA 安全性と比べて弱い安全性である “選択的タグ及び適応的選択暗号文攻撃に対する識別不可能性”(選択的タグ CCA 安全性, IND-stag-CCA) について述べる. 前節の定義にみられるように, IND-tag-CCA 安全性を攻撃する攻撃者は Phase 1. における復号クエリに対する応答を受け取ってからチャレンジタグを選ぶ事ができたが, IND-stag-CCA 安全性を攻撃する攻撃者は, Setup. においてチャレンジタグを宣言しなければならない. この意味で攻撃者にとって不利な条件下でのモデルであるため, 安全性はより弱いものとなる.

TBKEM TK の IND-stag-CCA は, 以下の攻撃者 $\mathcal{A}$ と IND-stag-CCA チャレンジャー $\mathcal{CH}$ 間の IND-stag-CCA ゲームを用いて定義される.

Setup. まず $\mathcal{CH}$ は $prm \leftarrow \mathsf{TSetup}(1^\kappa)$ を計算し, prm を $\mathcal{A}$ に渡す. 次に $\mathcal{A}$ は, チャレンジタグ tag^* を $\mathcal{CH}$ に送信する. 最後に $\mathcal{CH}$ は $(pk, sk) \leftarrow \mathsf{TKG}(prm)$ を計算し, 出力された pk を $\mathcal{A}$ に渡し, sk を保持しておく.

Phase 1. , Challenge. , Phase 2. , Guess. IND-tag-CCA ゲームと同様. ただし, Challenge. のときに $\mathcal{A}$ はチャレンジタグの出力は行わず, $\mathcal{CH}$ は Setup. の時に受け取ったタグ tag^* を用いて, チャレンジ暗号文を計算する.

ここで, ある TBKEM TK に対する $\mathcal{A}$ のアドバンテージを以下の様に定義する.

$$\mathsf{Adv}^{\text{IND-stag-CCA}}_{TK,\,\mathcal{A}} = |\Pr[b' = b] - \frac{1}{2}|$$

定義 2.3. 全ての多項式時間アルゴリズム $\mathcal{A}$ に対し, $\mathsf{Adv}^{\text{IND-stag-CCA}}_{TK,\,\mathcal{A}}$ が無視できる値であるとき, *TBKEM* TK は *IND-stag-CCA* 安全であるという.

TBKEM の構成　TBKEM はあまり一般的な構成要素ではないと思われるかもしれないが, Boyen ら [15, 16] の KEM や PKE の様な近年提案されたスタンダードモデルで安全性を証明できる KEM, PKE の多くは, 構成要素として衝突困難ハッシュ関数を用いているが, その内部のハッシュ関数にタグ tag を入力するだけで IND-stag-CCA 安全な KEM か, IND-tag-CCA 安全な TBKEM となることが, [36] において指摘されている. 前者の安全性を持つものに [16] の KEM, [30] の KEM, [25] などがあり, 後者の安全性を持つものに [15] の PKE から構成される TBKEM がある.

また, 全ての IND-CCA 安全な Tag-KEM も IND-tag-CCA 安全な TBKEM となることも知られている. これにより数多くの Tag-KEM の実際の構成が TBKEM として使われることが可能である (例えば [4, 3] などがあげられる). さらに, Abe ら [4] は IND-CCA 安全な Tag-KEM は, 任意の IND-CCA 安全な KEM と sUF-OT 安全な MAC(後述) から一般的に構成できる事が示されているため, IND-tag-CCA 安全な TBKEM もそれらの構成法を使う事ができる.

2.3　データカプセル化メカニズム (DEM)

データカプセル化メカニズム (Data Encapsulation Mechanism. 以下, DEM と表記する) とは, 公開鍵暗号と共通鍵暗号の長所を組み合わせた暗号方式であるハイブリッド暗号のうち, 乱数を含まない共通鍵暗号の部分を定式化したものである.

DEM は以下の 2 つのアルゴリズムから成る.

DEnc: 鍵 $K \in \mathcal{K}$, 平文 m を入力とし, 暗号文である c を出力する.

DDec: K, c を入力として m を出力する.

全ての K と m に対し, $m = \mathsf{DDec}(K, \mathsf{DEnc}(K, m))$ が求められる.

2.3.1　IND-CCA 安全性, IND-OT 安全性

本稿では, DEM の安全性として “適応的選択暗号文攻撃に対する識別不可能性” (IND-CCA), 及び “一回攻撃に対する識別不可能性” (IND-OT) を取り扱う. DEM D の IND-CCA は, 以下の攻撃者 $\mathcal{A}$ と IND-CCA チャレンジャー $\mathcal{CH}$ 間の IND-CCA ゲームを用いて定義される.

Setup. $\mathcal{CH}$ は鍵空間 $\mathcal{K}$ からランダムに K を選ぶ.

Phase 1. $\mathcal{A}$ は $\mathcal{CH}$ に対し, 任意の回数, 復号クエリ c を発行することができる. $\mathcal{CH}$ はそれぞれのクエリ c に対し, 正しい復号結果 $m \leftarrow \mathsf{DDec}(K, c)$ を返す.

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

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

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

ここで, ある DEM D に対する $\mathcal{A}$ のアドバンテージを以下の様に定義する.

$$\mathsf{Adv}^{\text{IND-CCA}}_{D,\,\mathcal{A}} = |\Pr[b' = b] - \frac{1}{2}|$$

加えて, 一回攻撃に対する識別不可能性 (IND-OT) を, IND-OT ゲームによって定義する. IND-OT ゲームは攻撃者が復号クエリを送信する事が出来ないという点以外, IND-CCA ゲームと同じである. 同様にして IND-OT アドバンテージ ($\mathsf{Adv}^{\text{IND-OT}}_{D,\,\mathcal{A}}$) を定義する.

定義 2.4. 全ての多項式時間アルゴリズム $\mathcal{A}$ に対し, $\mathsf{Adv}^{\text{IND-CCA}}_{D,\,\mathcal{A}}$ *(*同様に, $\mathsf{Adv}^{\text{IND-OT}}_{D,\,\mathcal{A}}$*)*が無視できる値であるとき, *DEM* D は *IND-CCA(*同様に, *IND-OT)* 安全であるという.

2.3.2　1 対 1 対応

DEM の性質の一つに, “1 対 1 対応” がある. ある DEM アルゴリズムが 1 対 1 対応であるとは, 鍵 K と平文 m が与えられたら $\mathsf{DDec}(K, c) = m$ を満たすような c は高々1 つしか存在しない事をさす. この性質は強擬似ランダム置換に基づく IND-CCA 安全な DEM [38] ならば満たされる. また, MAC が 1 対 1 対応を持つならば, [38] や有名な Encrypt-then-MAC による DEM の構成もまた 1 対 1 対応の性質を持つ [9] (MAC の 1 対 1 対応の定義については後述する).

2.4　電子署名

これまでに述べた暗号技術は情報の秘匿性についての安全性を達成するものであった. 本節では情報の完全性, 偽造不可能性を保守する暗号技術である電子署名について述べる.

電子署名は以下の 4 つのアルゴリズムからなる.

SSetup: セキュリティパラメータ 1^κ を入力とし, 公開パラメータ prm を出力する.

SKG: 公開パラメータ prm を入力とし, 公開鍵である暗号化鍵 sk と秘密鍵である検証鍵 vk の対を出力する.

Sign: prm, prm, sk, 平文 m を入力とし, 署名 σ を出力する.

SVer: prm, vk, m, σ を入力として $\top$ または $\bot$ を出力する.

全ての $prm \leftarrow \mathsf{SSetup}(1^\kappa)$, $(vk, sk) \leftarrow \mathsf{SKG}(prm)$, m に対し, $\mathsf{SVer}(prm, vk, m, \mathsf{Sign}(prm, sk, m)) = \top$ が求められる.

2.4.1　sUF-CMA 安全性

本稿では, 電子署名の安全性として “選択メッセージ攻撃に対する強偽造不可能性” (sUF-CMA) を取り扱う. 電子署名 S の sUF-CMA は, 以下の攻撃者 $\mathcal{A}$ と sUF-CMA チャレンジャー $\mathcal{CH}$ 間の sUF-CMA ゲームを用いて定義される.

Setup. $\mathcal{CH}$ は $prm \leftarrow \mathsf{SSetup}(1^\kappa)$, $(sk, vk) \leftarrow \mathsf{SKG}(prm)$ を計算する. 出力された (prm, vk) を $\mathcal{A}$ に渡し, sk を保持しておく.

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

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

ここで, ある電子署名方式 S に対する $\mathcal{A}$ のアドバンテージを以下のように定義する.

$$\mathsf{Adv}^{\text{sUF-CMA}}_{S,\,\mathcal{A}} = \Pr[\mathsf{SVer}(m^*, \sigma^*, vk) = \top \wedge (m^*, \sigma^*) \notin \mathcal{L}]$$

ここで $\mathcal{L}$ は, $\mathcal{A}$ が発行した全ての平文とそれに対する署名の組合せ, すなわち $\{(m_1, \sigma_1), ..., (m_q, \sigma_q)\}$ を表す.

定義 2.5. 全ての多項式時間アルゴリズム $\mathcal{A}$ に対し, $\mathsf{Adv}^{\text{sUF-CMA}}_{S,\,\mathcal{A}}$ が無視できる値である時, 電子署名 S は *sUF-CMA* 安全であるという.

2.5　メッセージ認証子 (MAC)

電子署名が公開鍵系の完全性, 偽造不可能性を満たす暗号技術であるのに対して, 共通鍵系の完全性, 偽造不可能性を満たす暗号技術にメッセージ認証子 (Message Authentication Code, 以下 MAC と表記する) がある.

MAC は以下の 2 つのアルゴリズムから成る.

Mac: 鍵 $K \in \mathcal{K}$, 平文 m を入力とし, メッセージ認証子である τ を出力する.

MVer: K, m, τ を入力として $\top$ または $\bot$ を出力する.

全ての K と m に対し, $\mathsf{MVer}(K, m, \mathsf{Mac}(K, m)) = \top$ が求められる.

2.5.1　sUF-OT 安全性

本稿では, MAC の安全性として “一回攻撃に対する偽造不可能性”(sUF-OT) を取り扱う. MAC M の sUF-OT は, 以下の攻撃者 $\mathcal{A}$ と sUF-OT チャレンジャー $\mathcal{CH}$ 間の sUF-OT ゲームを用いて定義される.

Setup. $\mathcal{CH}$ は鍵空間 $\mathcal{K}$ からランダムに K を選ぶ.

Query. $\mathcal{A}$ は $\mathcal{CH}$ に対し, 1 回だけクエリ m を発行することができる. $\mathcal{CH}$ は m に対し, 正しい MAC 値 $\tau \leftarrow \mathsf{Mac}(K, m)$ を返す.

Guess. $\mathcal{A}$ は平文と MAC 値の組 (m^*, τ^*) を出力する.

ここで, ある MAC M に対する $\mathcal{A}$ のアドバンテージを以下のように定義する.

$$\mathsf{Adv}^{\text{sUF-OT}}_{M,\,\mathcal{A}} = \Pr[\mathsf{MVer}(m^*, \tau^*, K) = \top \wedge (m^*, \tau^*) \neq (m, \tau)]$$

定義 2.6. 全ての多項式時間アルゴリズム $\mathcal{A}$ に対し, $\mathsf{Adv}^{\text{sUF-OT}}_{M,\,\mathcal{A}}$ が無視できる値である時, ある *MAC* M は *sUF-OT* 安全であるという.

2.5.2　1 対 1 対応

MAC の性質の一つに, “1 対 1 対応” がある. ある MAC アルゴリズムが 1 対 1 対応であるとは, 共通鍵 K と平文 M が与えられたとき, $\mathsf{MVer}(K, m, \tau) = \top$ を満たすような MAC タグ τ が高々1 つしかない事を指す. この性質は Mac アルゴリズムが確定的で, MVer アルゴリズムで共通鍵と与えられた平文から再計算を行って与えられた MAC タグと比較する構成であれば満たされる. 既存の大部分の MAC はこれにあてはまる.

2.6　カメレオンハッシュ

カメレオンハッシュとは, 同じハッシュ値を出力するメッセージと乱数の対を得るためのトラップドア付きのハッシュ関数である. カメレオンハッシュは以下の 3 つのアルゴリズムから成る.

HKG: セキュリティパラメータ 1^κ を入力とし, ハッシュ鍵/トラップドア対 (hk, td) を出力する.

CMH: ハッシュ鍵 hk, メッセージ x, 乱数 r を入力とし, ハッシュ値 y を出力する.

Switch: トラップドア td, メッセージ x', メッセージと乱数のペア (x, r) を入力として r' を出力する.

全ての $(hk, td) \leftarrow \mathsf{HKG}(1^\kappa)$, x, r, x', r' に対し, $\mathsf{CMH}(hk, x; r) = \mathsf{CMH}(hk, x'; r')$ が求められる. つまりトラップドア td は同じハッシュ値を出力するメッセージと乱数の対を得るための秘密鍵の役割を果たす.

2.6.1　カメレオンハッシュの安全性

カメレオンハッシュの安全性の定義として, 衝突困難性, Switch アルゴリズムにおける出力の一様性がある. この 2 つを満たす場合, カメレオンハッシュC は安全であるとする.

定義 2.7. 全ての多項式時間アルゴリズム $\mathcal{A}$ に対し, $\Pr[(hk, td) \leftarrow \mathsf{HKG}; ((x, r), (x', r')) \leftarrow \mathcal{A}(hk) : \mathsf{CMH}(hk, x; r) = \mathsf{CMH}(hk, x'; r') \wedge (x, r) \neq (x', r'))]$ が無視できる値であるとき, カメレオンハッシュC が衝突困難性を満たすという.

定義　2.8. 全ての m と m' について, r が一様ランダムに選ばれていれば $r' \leftarrow \mathsf{Switch}(td, (m, r), m')$ が一様ランダムに分布するとき, カメレオンハッシュC は Switch アルゴリズムにおける出力の一様性を持つという.

カメレオンハッシュの構成　有名な巡回群の離散対数問題の困難性に基づく方式を振り返る. g を巡回群 $\mathbb{G}$(位数 p) の生成元とする. HKG では, $a \in \mathbb{Z}_p^*$ を一様ランダムに選び, $h \leftarrow g^a$ を計算し, $(hk, td) = ((g, h), a)$ を出力する. $\mathsf{CMH}(hk, m; r)$ ($m \in \mathbb{Z}_p$, $r \in \mathbb{Z}_p$ は一様ランダムな値) では, $y \leftarrow g^m h^r$ を計算し出力する. $\mathsf{Switch}(td, (m, r), m')$ は $m' = m$ のときは r をそのまま, それ以外は $r' \leftarrow r + (m - m')/a$ を出力する.

Chapter 3 Signcryption

Signcryption は公開鍵暗号と電子署名を組み合わせた機能を持つ暗号技術である. Signcryption が保守する安全性は, 平文の秘匿性, メッセージの偽造不可能性, さらに否認防止である. 本章では Signcryption の詳細なアルゴリズムと安全性の定義について述べる.

3.1 Signcryption のアルゴリズム

Signcryption 方式では, 送信者は受信者の公開鍵を用いて送信したいメッセージを暗号化し, 同時に自身の秘密鍵を用いてメッセージに署名を発行する. 受信者は送られてきた情報を自身の秘密鍵を用いて復号し, 同時に送信者の公開鍵を用いて署名の検証を行う.

Signcryption は以下の 5 つのアルゴリズムから成る.

Setup: セキュリティパラメータ 1^κ を入力とし, 公開パラメータ prm を出力する.

$\mathsf{KeyGen_R}$: prm を入力とし, 受信者用の公開鍵/秘密鍵対 (pk_R, sk_R) を出力する.

$\mathsf{KeyGen_S}$: prm を入力とし, 送信者用の公開鍵/秘密鍵対 (pk_S, sk_S) を出力する.

SC: 暗号化と署名を行うアルゴリズム. prm, pk_R, sk_S, 平文 m を入力とし, 暗号文 c を出力する.

USC: 復号と検証を行うアルゴリズム. prm, sk_R, pk_S, c を入力とし, 平文 m(あるいは復号失敗 "$\perp$") を出力する.

全ての $prm \leftarrow \mathsf{Setup}(1^\kappa)$, $(pk_R, sk_R) \leftarrow \mathsf{KeyGen_R}(prm)$, m, $(pk_S, sk_S) \leftarrow \mathsf{KeyGen_S}(prm)$, $c \leftarrow \mathsf{SC}(prm, sk_S, pk_R, m)$ に対し, $m = \mathsf{USC}(prm, pk_S, sk_R, c)$ が求められる.

3.2 Signcryption の安全性

これまで定義されてきた Signcryption の安全性のうち, 最も強い証明可能安全性を考える. ここでは, 一人の送信者と一人の受信者からなる二人モデルではなく, 多人数でのモデルを考える. 通常の公開鍵暗号や署名の場合とは異なり, Signcryption では二人モデルでの安全性は多人数モデルでの安全性を包含しないため, Signcryption の安全性モデルとして多人数モデルを考える事は非常に重要である.

さらに, 本稿ではその中でも強いモデルである内部者に対する安全性を考える. 内部者とは, 攻撃対象の送信者から受信者のうち, 片方の秘密鍵を知ることができる攻撃者である. これらは [36] で詳細に定義されている.

ここで, 秘匿性と偽造不可能性それぞれについての定義を振り返る ([36] の表記法を用いる). より具体的には,

- 秘匿性については, 動的多人数モデルにおける内部者の選択暗号文攻撃に対する識別不可能性 (dM-IND-iCCA) を扱う
- 偽造不可能性については, 動的多人数モデルにおける内部者の選択メッセージ攻撃に対する強偽造不可能性 (dM-sUF-iCMA) を扱う

本節では, これら 2 種の安全性について詳細に定義する.

3.2.1 dM-IND-iCCA 安全性

内部者で秘匿性を攻撃する攻撃者を考える場合, 攻撃者が送信者/受信者の公開鍵だけでなく送信者の秘密鍵 sk_S も知っている状況を想定する. Signcryption 方式 SC の dM-IND-iCCA は, 以下の攻撃者 $\mathcal{A}$ と dM-IND-iCCA チャレンジャー $\mathcal{CH}$ 間の dM-IND-iCCA ゲームを用いて定義される.

Setup. $\mathcal{CH}$ は $prm \leftarrow \mathsf{Setup}(1^\kappa)$, 及び $(pk_R, sk_R) \leftarrow \mathsf{KeyGen_R}(prm)$ を計算し, 出力された (prm, pk_R) を $\mathcal{A}$ に渡し, sk_R を保持しておく.

Phase 1. $\mathcal{A}$ は $\mathcal{CH}$ に対し, 任意の回数, 復号クエリ (pk_S, c) を発行することができる. $\mathcal{CH}$ はそれぞれのクエリ (pk_S, c) に対し, 正しい復号結果 $m/\perp \leftarrow \mathsf{USC}(prm, pk_S, sk_R, c)$ を返す.

Challenge. $\mathcal{A}$ は 2 つの任意の平文 m_0, m_1, および任意の送信者の (pk_S^*, sk_S^*) を選び, $\mathcal{CH}$ に送る. $\mathcal{CH}$ はランダムにコイン $b \in \{0,1\}$ を振り, m_b の暗号文 $c^* \leftarrow \mathsf{SC}(prm, sk_S^*, pk_R, m_b)$ を計算し, c^* を $\mathcal{A}$ に渡す.

Phase 2. $\mathcal{A}$ は Phase 1. と同様に復号クエリを発行することができる. ただし, $\mathcal{A}$ は (pk_S^*, c^*) を復号クエリとすることはできない.

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

ここで, ある Signcryption 方式 SC に対する $\mathcal{A}$ のアドバンテージを以下の様に定義する.

$$\mathsf{Adv}^{\text{dM-IND-iCCA}}_{SC,\,\mathcal{A}} = |\Pr[b' = b] - \frac{1}{2}|$$

定義 3.1. 全ての多項式時間アルゴリズム $\mathcal{A}$ に対し, $\mathsf{Adv}^{\text{dM-IND-iCCA}}_{SC,\,\mathcal{A}}$ が無視できる値であるとき, *Signcryption* 方式 SC は *dM-IND-iCCA* 安全であるという.

3.2.2　dM-sUF-iCMA 安全性

内部者で偽造不可能性を攻撃する攻撃者を考える場合, 攻撃者が送信者/受信者の公開鍵だけでなく受信者の秘密鍵 sk_R も知っている状況を想定する. Signcryption 方式 SC の dM-sUF-iCMA は, 以下の攻撃者 $\mathcal{A}$ と dM-sUF-iCMA チャレンジャー $\mathcal{CH}$ 間の dM-sUF-iCMA ゲームを用いて定義される.

Setup. $\mathcal{CH}$ は $prm \leftarrow \mathsf{Setup}(1^\kappa)$, 及び $(pk_S, sk_S) \leftarrow \mathsf{KeyGen_S}(prm)$ を計算し, 出力された (prm, pk_S) を $\mathcal{A}$ に渡し, sk_S を保持しておく.

Query. $\mathcal{A}$ は $\mathcal{CH}$ に対し, 暗号化クエリ (pk_R, m) を発行することができる. $\mathcal{CH}$ は (pk_R, m) に対し, 正しい暗号文 $c \leftarrow \mathsf{SC}(prm, pk_R, sk_S, m)$ を返す.

Guess. $\mathcal{A}$ は任意の受信者の公開鍵/秘密鍵, 及び暗号文の組合せ (pk_R^*, sk_R^*, c^*) を出力する.

ここで, ある Signcryption 方式 SC に対する $\mathcal{A}$ のアドバンテージを以下のように定義する.

$$\mathsf{Adv}^{\text{dM-sUF-iCMA}}_{SC,\ \mathcal{A}} = \Pr[\mathsf{USC}(prm, c^*, sk_R^*, pk_S^*) = m^* \neq \perp \wedge (pk_R^*, c^*, m^*) \notin \mathcal{L}]$$

ここで $\mathcal{L}$ は, $\mathcal{A}$ が発行した全ての平文とそれに対する署名の組合せ, すなわち $\{(pk_{R1}, c_1), ..., (pk_{Rq}, c_q)\}$ を表す.

定義 3.2. 全ての多項式時間アルゴリズム $\mathcal{A}$ に対し, $\mathsf{Adv}^{\text{dM-sUF-iCMA}}_{SC,\ \mathcal{A}}$ が無視できる値である時, *Signcryption* 方式 SC は *dM-sUF-iCMA* 安全であるという.

Chapter 4　関連研究

本章では本研究の関連研究を紹介する. ここでは, 提案された Signcryption の構成法が一般的構成法によるものか否かによって分けて述べる事とする (一般的構成法については 1.3 節を参照されたい).

4.1　Signcryption に関する既存研究

Signcryption は 1997 年に Zheng [45] によってはじめてその概念が提案された. [45] の主たる動機は, これらの安全性を満たした通信を行う場合に, 単純に電子署名と公開鍵暗号を組み合わせるよりも Signcryption を利用した方が遥かに効率が良い, という点にあった. [45] で提案された構成は正式な安全性証明がなされなかったが, これは後の [7, 8] において行われた.

この提案以降, いくつかの異なる安全性モデルのもと, 多くの Signcryption の構成法が提案されてきた ([45, 6, 7, 32, 33, 20, 8, 31, 43, 36]). 最もシンプルな Signcryption の安全性モデルは一人の送信者と一人の受信者からなるいわゆる二人モデルである (このモデルは [6, 20] のような初期の論文で採用されている). しかし Dent [20] は, 通常の公開鍵暗号や電子署名とは異なり Signcryption では二人モデルでの安全性は多人数モデルでの安全性を包含しない事を示した. 従って Signcryption の安全性モデルとして多人数モデルを考える事は非常に重要である. また, もう一つの Signcryption の安全性定義の重要な要素として, 攻撃者が送信者や受信者を攻撃する外部者なのか, 送信者または受信者の役割の一部を担う内部者なのか, という点がある. 外部者安全性より内部者安全性の方が強いため, Signcryption が内部者安全性を達成する方が望ましい. 多人数モデルにおける最も強い安全性定義 (秘匿性と偽造不可能性) は [32] で定義され, 使われている.

これまで提案された多くの手法は上記の安全性を混合して証明が行われてきた. [8] では内部者に対する秘匿性と外部者に対する偽造不可能性を達成する手法が提案され, [24, 28] では外部者に対する秘匿性と内部者に対する偽造不可能性を達成する手法が提案された. 最も強い意味での安全性, すなわち内部者に対する秘匿性と偽造不可能性を証明できる手法を構成することは非常に困難な課題であった. 例えば, [32] で提案された手法は [41] によって安全性が破られ, [34] で提案された手法は [42] によって破られた. そして Libert ら [33] は彼ら自身の手法を改良して [41] による攻撃から防ぐ手法を提案した. このように内部者安全の Signcryption 方式を提案する際は注意が必要であるとわかる.

ランダムオラクルを許すならば, 多人数モデルで内部者に対する秘匿性と強偽造不可能性を達成する Signcryption がこれまでにいくつか提案されている ([33, 31, 36]). しかしランダムオラクルはいくつかの問題点がある事が知られているため [18](詳細は

1.2 節を参照されたい), スタンダードモデルでの安全性を考える事は有用である.

多人数モデルで内部者安全性をスタンダードモデルで達成する Signcryption もまたいくつか提案されている [6, 43, 36]. Tan [43] は最も強い意味での内部者に対する秘匿性を達成したが, 彼の手法の内部者に対する強偽造不可能性は “鍵登録” (詳細は 4.2 節を参照されたい) と呼ばれる制限されたモデルでのみ有効である.

4.2　Signcryption の一般的構成法の既存研究

Signcryption の一般的構成法として, An らによる研究と, Matsuda らによる研究の 2 つを紹介する.

4.2.1　An らによる研究

An ら [6] は公開鍵暗号と電子署名を単純に組み合わせた Signcryption の一般的構成法を構成し, 安全性についての分析を行った. 彼らは伝統的な 2 人モデルでの Sign-then-Encrypt と Encrypt-then-Sign を多人数モデルでの安全性に強める手法を示した. しかし彼らの手法は通常の選択暗号文攻撃に対する安全性よりも弱いいわゆる *generalized chosen ciphertext security* のみを満たすものであり, 一般的に適応可能とは言い難いものであった.

4.2.2　Matsuda らによる研究

Signcryption の一般的構成法のうち, 達成される安全性が強く最も効率のよいものとして, Matsuda ら [36] が提案した一般的構成法について詳しく紹介する.

彼らの手法では構成要素としてタグベース暗号 (TBE) やタグベース KEM(TBKEM) を用いている. 具体的には, TBKEM と DEM の組み合わせを TBE と見て, 署名方式と合わせて Sign-then-Encrypt 方式の Signcryption とする. 彼らの一般的構成法は強い意味での内部者に対する秘匿性を達成するが, 偽造不可能性に関しては内部者に対する “弱” 偽造不可能性しか達成していない. ただし, 鍵登録と呼ばれるモデルでは内部者に対する強偽造不可能性を達成している. 以下に鍵登録について述べる.

鍵登録の必要性　鍵登録の必要性とは, 攻撃者がある暗号方式を攻撃する際, 利用する公開鍵に対応する秘密鍵を全て公開する事が要求される, という仮定である. 実際にこの仮定を実現するには, 伝統的な公開鍵暗号基盤である PKI が利用可能であり, かつ全てのパーティが認証局とゼロ知識証明 (各パーティの公開鍵についての証明書を入手する前にそれらの秘密鍵を知っている事の証明) ができる環境にある事を必要とする. しかし, これらの証明を実行することは認証局に大きな負荷をかけることになるため, このような鍵登録は実用的なシステムでは用いられていない.

また, 彼らの構成法にはもう一つの仮定として “Signcryption 結合可能性” と呼ばれるものを必要とする. 以下に Signcryption 結合可能性について詳しく述べる.

Signcryption 結合可能性　TBKEM の暗号文 c と署名方式の署名 σ が,

(i) 乱数 “のみ” による部分 c_1(署名は σ_1) と “それ以外” の部分 c_2(署名は σ_2) に分割可能 (Partitionable) であり,
(ii) 互いに乱数の空間が共有でき, [1]
(iii) 乱数のみに依存した部分 c_1(署名の場合は σ_1) と秘密鍵が与えられれば, r の値を直接知らずに “それ以外” の部分 (KEM の場合は共通鍵 K も含む) という安全性証明の際に必要な Simulatability と呼ばれる性質を満たす[2]

ならば Signcryption としての暗号文を作る際に TBKEM の暗号文と署名方式の署名のうち, “乱数のみによる要素” を共有できるというものである. [36] では, TBKEM と署名が対になって上記性質を満たす場合, それら二つの方式は Signcryption 結合可能であると呼ばれた. また, 構成要素の TBKEM が IND-tag-CCA 安全, 署名方式が強偽造不可能, かつ DEM が IND-CCA 安全であれば, [36] 方式での最も強いモデルでの CCA 安全, かつ内部者に対し強偽造不可能な Signcryption 方式の構成が可能であることが示された. このうち, TBKEM が IND-stag-CCA 安全性しか満たさない場合, 秘匿性がやや弱いモデルでの安全性しか満たさないことも示された.

[36] の構成法は鍵登録や Signcryption 結合可能性といった特殊な仮定を必要としているが, これまでに提案された方式の中でも達成される安全性が強く, 最も効率がよいという事を強調したい. 我々の知る限り, 多人数モデルで (特殊な仮定なしに) 内部者に対する秘匿性と強偽造不可能性をスタンダードモデルで達成する Signcryption の構成法は存在しない. 本稿ではそのような Signcryption の構成を目指す.

[1]分割可能な TBKEM の要求を正式に書くと以下の二つの性質になる. (1) TEncap アルゴリズムが, 二つの決定的なアルゴリズム TE_1 と TE_2 に分割され, 乱数を r とすると, 全ての $(K, c) \leftarrow \mathsf{TEncap}(prm, pk, \mathsf{tag}; r)$ の計算が $c = (c_1, c_2)$ かつ $c_1 \leftarrow \mathsf{TE}_1(prm, r)$ 及び $(c_2, K) \leftarrow \mathsf{TE}_2(prm, pk, \mathsf{tag}, r)$ と書ける. (2) 全ての $prm \leftarrow \mathsf{TSetup}$, 全ての $(pk, sk) \leftarrow \mathsf{TKG}(prm)$, 及び全てのタグ tag, 全ての c_1 について, $\mathsf{TDecap}(prm, sk, \mathsf{tag}, (c_1, c_2)) = K$ となる (c_2, K) の組は高々一つしか存在しない.

[2]例えば, Boyen ら [16] の KEM は, $pk = (g, \hat{g}, X = g^x, Y = g^y, \mathsf{Z} = e(g, \hat{g})^\alpha)$, $sk = (x, y, \alpha)$, 乱数のみに依存する部分 $c_1 = g^r$, それ以外の部分 $c_2 = (XY^{H(c_1)})^r$, 共通鍵 $K = \mathsf{Z}^r$ という様に暗号文 $c = (c_1, c_2)$ を分割でき, $sk = (x, y, \alpha)$ が分かれば, c_1 を用いて, $c_2 = c_1^{x+y \cdot H(c_1)}$, $K = e(c_1, \hat{g})^\alpha$ を計算できる. この様にして作られた c_2 と K は, Boyen らの KEM 本来の方式の暗号化アルゴリズムを, $c_1 = g^r$ となる乱数 r を使って実行した場合と完全に同じ出力となる.

Chapter 5　TBKEMの強化手法

Kiltz [29] は, IND-CCA 安全な PKE を計算コスト増加の実質ゼロ, 暗号文サイズ増加はタグのサイズ分, あるいはタグのハッシュ値分で適応的タグ CCA 安全な TBE へと一般的に変換する方法を示した. 全ての (選択的) タグ CCA 安全な TBE は CCA 安全な PKE として当然使用可能であるため, 彼の方式では選択的タグ CCA 安全な TBE を適応的タグ CCA 安全性を持つ方式へとも変換できる.

では選択的タグ CCA 安全な TBKEM から適応的タグ CCA 安全な TBKEM への変換はどうだろうか. [36] では, Boyen ら [15, 16] の KEM や PKE の様な近年提案されたスタンダードモデルで CCA 安全性を証明できる KEM, PKE の多くは構成要素として衝突困難ハッシュ関数を用いているが, その内部のハッシュ関数にタグを入力するだけで選択的タグ CCA 安全性か, 適応的タグ CCA 安全性を持つ TBKEM となることが指摘されている. 前者の安全性を持つものに [16] の KEM, [30] の KEM, [25] など, これ以外にもいくつか効率的な方式が知られているのに対し, 後者の安全性を持つものには [15] の PKE, あるいは [30] の PKE しか無く, これらの方式はいわゆる “Waters のハッシュ” を構成要素として用いる必要があり, 公開鍵 (あるいは秘密鍵も) のサイズが莫大になるという欠点がある.

そこで, 効率的な選択的タグ CCA 安全な TBKEM から適応的タグ安全な TBKEM への効率的な安全性強化手法 (変換) を考えたい. 本章ではカメレオンハッシュを用いて選択的タグ CCA 安全な TBKEM を適応的タグ CCA 安全な TBKEM への強化手法を提案する. 既に述べた Kiltz [29] の PKE から TBE への変換を用いる様な自明な同目的を達成する自明な手法が存在するが, 本手法は自明な手法と比べ変換後の暗号文サイズを小さくできるため, 変換後の暗号文サイズの増加を抑えたい場合には本提案手法は便利である.

また, 本稿での提案手法の応用先として, スタンダードモデルで “内部者” に対する安全性を証明可能な Signcryption の構成を考える. 最近 [36] において TBKEM のうち特殊な性質を持つものを構成要素の一つとする Signcryption の構成法が提案された. 彼らの構成法では, TBKEM が適応的タグ CCA 安全ならば最も強い秘匿性を, 選択的タグ CCA 安全であればそれよりも弱いモデルでの安全性を達成できることが示されている. これに対し本強化手法では彼らの Signcryption 方式で TBKEM に必要とされる “分割可能性” という性質と, ある電子署名との TBKEM との間で考えられる性質である “Signcryption 結合可能性” と呼ばれる性質が保存され, 変換後の TBKEM とともに [36] の Signcryption 結合可能性を持つ署名方式が存在する. 従って, 既に [36] で示された多くの選択的タグ CCA 安全な TBKEM に我々の変換を適用すれば, 今までよりもより多くの強いモデルでの安全性をスタンダードモデルで証明可能な Signcryption 方式が構成できる.

さらに, 応用先として考えられる Signcryption の構成は [36] の手法だけではなく,

$\mathsf{TKG}'(1^\kappa)$:
　$(pk, sk) \leftarrow \mathsf{TKG}(1^\kappa)$
　$(hk, td) \leftarrow \mathsf{HKG}(1^\kappa)$
　$SK \leftarrow sk$, $PK \leftarrow (pk, hk)$
　Output (SK, PK).

$\mathsf{TEncap}'(PK, \mathsf{tag})$:
　$r \leftarrow \mathcal{R}$
　$\mathsf{tag}' \leftarrow \mathsf{CMH}(hk, \mathsf{tag}; r)$
　$(c, K) \leftarrow \mathsf{TEncap}(pk, \mathsf{tag}')$
　$c' \leftarrow (c, r)$
　Output (c', K).

$\mathsf{TDecap}'(SK, \mathsf{tag}, c')$:
　$(c, r) \leftarrow c'$
　$\mathsf{tag}' \leftarrow \mathsf{CMH}(hk, \mathsf{tag}; r)$
　Output $K \leftarrow \mathsf{TDecap}(sk, \mathsf{tag}', c)$

図 5.1: 新しく提案する TBKEM のアルゴリズム TK' ($\mathsf{TSetup}' = \mathsf{TSetup}$ とする)

IND-tag-CCA 安全な TBKEM を構成要素とする一般的構成法であればどのような Signcryption の構成法にも適用可能である. 本稿におけるもう一つの研究成果である Signcryption の一般的構成法は構成要素として IND-tag-CCA 安全な TBKEM を要求するため, この強化手法が利用可能である.

5.1　提案手法

TK : ($\mathsf{TSetup}, \mathsf{TKG}, \mathsf{TEncap}, \mathsf{TDecap}$) を IND-stag-CCA 安全な TBKEM とする. また, C : ($\mathsf{HKG}, \mathsf{CMH}, \mathsf{Switch}$) を安全なカメレオンハッシュとする. このとき TBKEM のアルゴリズム TK' を図 5.1 の様に構成する ($\mathcal{R}$ は C の乱数空間である).

構成のアイデア　2.2 節で述べた様に, 選択的タグ CCA 安全な TBKEM から適応的タグ CCA 安全な TBKEM に変換するには, Phase 1. における復号クエリに対する応答を受け取ってからチャレンジタグを選ぶ事ができるという, 攻撃者に有利な条件のもと CCA 安全性を満たさなければならない. そこで提案手法では, “仮の” タグをあらかじめ準備しておき Challenge. の際に $\mathcal{CH}$ には区別のつかない別のタグに差し替えるために, カメレオンハッシュを導入する.

5.2　安全性証明

定理 5.1. 提案手法で変換したい *TBKEM* TK が *IND-stag-CCA* 安全, かつ C が安全なカメレオンハッシュならば, 変換後の *TBKEM* である TK' は *IND-tag-CCA* の安全性を満たす.

定理 5.1 の証明　定理の証明にあたり, IND-tag-CCA ゲームに $1/2 + \mathsf{Adv}^{\text{IND-tag-CCA}}_{TK',\,\mathcal{A}}$ の確率で勝利する攻撃者 $\mathcal{A}$ が存在すると仮定する. 矛盾を導くために, $\mathsf{Adv}^{\text{IND-tag-CCA}}_{TK',\,\mathcal{A}}$ を無視できないと仮定する. また, $\mathcal{A}$ を構成要素として用いて TBKEM TK の IND-stag-CCA を多項式時間で破るシミュレータ $\mathcal{S}$ を構成する. $\mathcal{S}$ は $\mathcal{A}$ に対して以下のように IND-tag-CCA ゲームのシミュレートを行いつつ利用して, 自身の TK についての IND-stag-CCA ゲームを行う (アスタリスク (*) のついた値は Challenge. の際に用いられるとする).

Setup. prm を受け取り, $\mathcal{S}$ は $(hk, td) \leftarrow \mathsf{HKG}(1^\kappa)$ を計算する. 次に, ランダムに生成した $\overline{\mathsf{tag}}$ と乱数 $\bar{r}$ を入力として $\mathsf{tag}^*_{\mathcal{S}} \leftarrow \mathsf{CMH}(hk, \overline{\mathsf{tag}}; \bar{r})$ を計算し, $\mathcal{CH}$ に $\mathsf{tag}^*_{\mathcal{S}}$ をチャレンジタグとして宣言する. 最後に, $\mathcal{S}$ は $\mathcal{CH}$ から pk を受け取り, (prm, pk, hk) を入力として $\mathcal{A}$ を起動する.

Phase 1. $\mathcal{A}$ の発行する復号クエリ $(\mathsf{tag}_{\mathcal{A}}, c')$ に対して, $\mathcal{S}$ はまず $(c, r) = c'$ を用いて $\mathsf{tag}_{\mathcal{S}} \leftarrow \mathsf{CMH}(hk, \mathsf{tag}_{\mathcal{A}}; r)$ を計算する. 次に, $(\mathsf{tag}_{\mathcal{S}}, c)$ を復号クエリとして $\mathcal{CH}$ に送信し, 返された値 $K/\perp$ を $\mathcal{A}$ に返す.

Challenge. $\mathcal{A}$ が $\mathsf{tag}^*_{\mathcal{A}}$ を出力したとき, $\mathcal{S}$ は以下のように応答する. まず, チャレンジ暗号文/鍵対のリクエストを $\mathcal{CH}$ に送信し, (c^*, K^*_b) を受け取る. 次に, $r^* \leftarrow \mathsf{Switch}(td, (\overline{\mathsf{tag}}, \bar{r}), \mathsf{tag}^*_{\mathcal{A}})$ を実行する. 最後に $c'^* = (c^*, r^*)$ として, (c'^*, K^*_b) を $\mathcal{A}$ に返す.

Phase 2. $\mathcal{A}$ の発行する復号クエリ $(\mathsf{tag}_{\mathcal{A}}, c' = (c, r))$ について $\mathcal{S}$ は以下のように応答する. まず, Phase 1. と同様にして $\mathsf{tag}_{\mathcal{S}} \leftarrow \mathsf{CMH}(hk, \mathsf{tag}_{\mathcal{A}}; r)$ の計算を行う.

1. $(\mathsf{tag}_{\mathcal{S}}, c) = (\mathsf{tag}^*_{\mathcal{S}}, c^*)$ **である場合**
   $\mathcal{S}$ はシミュレーションを諦め, 異常終了する.
2. **それ以外の場合**
   Phase 1. と同様の動作を行い, 復号結果 $K/\perp$ を返す.

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

以上が $\mathcal{S}$ の構成である. $\mathcal{S}$ が禁止クエリである (tag^*, c^*) を $\mathcal{CH}$ に発行していない事, 並びに $\mathcal{A}$ にとってのチャレンジ暗号文は, カメレオンハッシュ C の持つ Switch アルゴリズムの出力の一様性によって正しく分布していることに注目されたい.

ここで, 以下のように 2 つのイベントを定義する.

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

Bad: Phase 2. において $\mathcal{A}$ が復号クエリ $(\mathsf{tag}_{\mathcal{A}}, (c, r))$ に, $\mathsf{tag}_{\mathcal{S}} = \mathsf{CMH}(hk, \mathsf{tag}_{\mathcal{A}}; r)$ として $(\mathsf{tag}_{\mathcal{S}}, c) = (\mathsf{tag}^*_{\mathcal{S}}, c^*)$ となるクエリを少なくとも一回は出す (このとき,$\mathcal{S}$ の構成より $\mathcal{S}$ は異常終了する).

ここで, $\mathcal{S}$ が IND-stag-CCA ゲームに勝利する確率を計算する. すると,

$$\begin{aligned}\Pr[\mathsf{Succ}] &\geq \Pr[\mathsf{Succ} \wedge \overline{\mathsf{Bad}}] \\ &= \Pr[\mathsf{Succ}|\overline{\mathsf{Bad}}] \cdot (1 - \Pr[\mathsf{Bad}]) \\ &\geq \Pr[\mathsf{Succ}|\overline{\mathsf{Bad}}] - \Pr[\mathsf{Bad}] \qquad (5.1)\end{aligned}$$

が得られる.

証明を完了するために以下の 2 つの補題を示す.

補題 5.1. $\Pr[\mathsf{Succ}|\overline{\mathsf{Bad}}] = 1/2 + \mathsf{Adv}^{\text{IND-tag-CCA}}_{TK,\,\mathcal{A}}$

補題 5.1 の証明　Bad が起こらないとき, $\mathcal{S}$ は $\mathcal{A}$ に対して IND-tag-CCA のゲームを完全にシミュレートしているため, 仮定より $\mathcal{S}$ は IND-stag-CCA ゲームに, $1/2 + \mathsf{Adv}^{\text{IND-stag-CCA}}_{TK,\,\mathcal{A}}$ の確率で勝利する. □

補題 5.2. $\Pr[\mathsf{Bad}]$ は無視できる値である.

補題 5.2 の証明　矛盾を導くために, $\Pr[\mathsf{Bad}]$ を無視できないと仮定する. また, $\mathcal{A}$ を構成要素として用いてカメレオンハッシュ C の衝突困難性を多項式時間で破るシミュレータ $\mathcal{S}$ を構成する. $\mathcal{S}$ は $\mathcal{A}$ に対して以下のように IND-tag-CCA ゲームのシミュレートを行いつつ利用して, 自身の C についての衝突を出力する.

Setup. hk を受け取り, $\mathcal{S}$ は $prm \leftarrow \mathsf{TSetup}(1^\kappa)$, 及び $(pk, sk) \leftarrow \mathsf{TKG}(prm)$ を計算する. 次に $\mathcal{S}$ は, (pk, hk) を入力として $\mathcal{A}$ を起動する.

Phase 1. $\mathcal{S}$ は $\mathcal{A}$ の発行する復号クエリ $(tag_{\mathcal{A}}, c')$ に対して, $K/\perp \leftarrow \mathsf{TDecap}'(prm, sk, \mathsf{tag}_{\mathcal{A}}, c')$ を返す.

Challenge. $\mathcal{A}$ が $(\mathsf{tag}^*_{\mathcal{A}})$ を出力したとき, $\mathcal{S}$ は以下のように応答する. まず, 乱数 $r^* \in \mathcal{R}$ を一様ランダムに選び, $\mathsf{tag}^*_{\mathcal{S}'} \leftarrow \mathsf{CMH}(hk, \mathsf{tag}^*_{\mathcal{A}}; r)$ を計算する. 次に, $\mathsf{tag}^*_{\mathcal{S}}$ に対する暗号文/鍵対 $(c^*, K^*_1) \leftarrow \mathsf{TEncap}(prm, pk, tag^*)$ を計算する. さらに, ランダムにコイン $b \in \{0, 1\}$ を振り, $K^*_0 \in \mathcal{K}$ を一様ランダムに選ぶ. 最後に $c'^* = (c^*, r^*)$ として, (c'^*, K^*_b) を $\mathcal{A}$ に返す.

Phase 2. $\mathcal{A}$ の発行する復号クエリ $(\mathsf{tag}_{\mathcal{A}}, c' = (c, r))$ について $\mathcal{S}$ は以下のように応答する. まず, Phase 1. と同様にして $\mathsf{tag}_{\mathcal{S}} \leftarrow \mathsf{CMH}(hk, \mathsf{tag}_{\mathcal{A}}; r)$ の計算を行う.

1. $(\mathsf{tag}_{\mathcal{S}}, c) = (\mathsf{tag}^*_{\mathcal{S}}, c^*)$ **である場合**
   $(\mathsf{tag}_{\mathcal{A}}, r), (\mathsf{tag}^*_{\mathcal{A}}, r^*)$ の対を出力して終了.

2. **それ以外の場合**
   Phase 1. と同様の動作を行う.

Guess. $\mathcal{A}$ が止まるまで Bad が起きなければ, $\mathcal{S}$ は諦めて停止する.

$\mathcal{S}$ は $\mathcal{A}$ に対して IND-tag-CCA ゲームを完全にシミュレートしているため, $\mathcal{S}$ は, $\mathcal{A}$ が Bad となる復号クエリを出すときには必ずカメレオンハッシュの衝突困難性を破る. ただしこれは C が衝突困難性を持つ事に矛盾するため, Pr[Bad] は無視できる値となる. 以上より, 補題 5.2 が証明された. □

以上をまとめる. 不等式 (5.1), 補題 5.1, 補題 5.2 より,

$$\mathsf{Adv}^{\text{IND-stag-CCA}}_{TK,\,\mathcal{S}} = |\Pr[\mathsf{Succ}] - \frac{1}{2}| = \mathsf{Adv}^{\text{IND-tag-CCA}}_{TK',\,\mathcal{A}} - \Pr[\mathsf{Bad}]$$

は無視できない値である. これは, TK が IND-stag-CCA 安全性を持つ事に矛盾する. 以上より定理 5.1 が証明された. □

5.3　効率評価

本稿でいう, IND-stag-CCA 安全な TBKEM から IND-tag-CCA 安全な TBKEM への自明な強化手法とは, 以下に示す通りである. まず, 全ての IND-stag-CCA 安全な TBKEM は IND-CCA 安全な KEM(TBKEM ではない) として用いる事ができる. IND-CCA 安全な KEM は IND-CCA 安全な DEM と組み合わせれば IND-CCA 安全な PKE となる. Kiltz [29] がタグを平文の一部として IND-CCA 安全な PKE で暗号化することで, IND-tag-CCA 安全なタグベース暗号 (TBE) とできる事を示した. IND-tag-CCA 安全な TBE は一様ランダムに選んだ鍵 K を暗号化することで IND-tag-CCA 安全な TBKEM として用いる事ができる. この方式の利点は変換の際に計算オーバーヘッドが発生しない事であるが, セキュリティパラメータを κ としたときに κ ビット, 及びタグを平文として暗号化する分, タグのサイズの暗号文サイズのオーバーヘッドが生じる. タグに関するサイズのオーバーヘッドは, 衝突困難ハッシュを用いることで 2κ ビット程度とできるが, 少なくとも 3κ ビットのオーバーヘッドを生じる.

これに対し, 5.1 節で示した提案手法のようにカメレオンハッシュを用いれば, 暗号文サイズに関しては乱数 r の長さのオーバーヘッドが生じ, r はおよそ 2κ ビットとできる. ただし, カメレオンハッシュの計算のため, 暗号化及び復号の際に一回の多重累乗計算が必要となる.

Chapter 6　多人数モデルで内部者安全な Signcryption

本章では多人数モデルで内部者に対する秘匿性と強偽造不可能性を達成する Signcryption の一般的構成法を 2 つ提案する. 1 つ目の構成法は TBKEM, DEM, 署名を用い, 2 つ目の構成法は KEM, DEM, 署名, MAC を用いる.

多人数モデルでも安全性を得るために, 我々は [36] と同様のアイデアを用いる. すなわち, 受信者の公開鍵を署名し, かつ送信者用の公開鍵をタグベース要素技術のタグとして用いる. 過去の方式と異なる点は我々の方式がランダムオラクルを用いずに内部者に対する強偽造不可能性を達成できる点である.

6.1　TBKEM を用いる一般的構成法

多人数モデルで内部者安全な Signcryption の一般的構成法を TBKEM などの要素技術を用いて構成する手法を示す.

6.1.1　提案手法

TK : (TSetup, TKG, TEncap, TDecap) を IND-tag-CCA 安全な TBKEM とする. また, D : (DEnc, DDec) を IND-CCA 安全な DEM とし, S : (SSetup, SKG, Sign, SVer) を強偽造不可な署名とする. このとき Signcryption 方式 SC_{tk} を図 6.1 の様に構成する. 一般性を失わずに, TK 及び D の鍵空間を $\{0,1\}^{\kappa}$ とする.

構成のアイデア　提案手法は Sign-then-Encrypt の一種とみなせる. 受信者の秘密鍵を持ち Signcryption クエリによって暗号文を受信できる内部攻撃者は, 暗号文を復号して再び暗号化できる (強偽造不可能性という意味では偽造に成功してしまう) ため, 通常の Sign-then-Encrypt では (強ではなく) 弱偽造不可能性しか達成できない. しかし提案手法では KEM/DEM による構成法を用いており, 平文と同時に KEM の暗号文を署名する事ができる. これにより, 内部の攻撃者は上記の攻撃を行う事ができず, 強偽造不可能性を破るために DEM の暗号文を修正する必要がある. しかし DEM が 1 対 1 対応であればそのような修正からもうまく防ぐ事ができる. 2.3 節で述べたように, 1 対 1 対応の DEM は容易に構成できる.

6.1.2　スタンダードモデルでの安全性証明

SC_{tk} の安全性は以下の 2 つの定理によって保証されている.

$\mathsf{Setup}(1^\kappa)$: $prm_{tk} \leftarrow \mathsf{TSetup}(1^\kappa)$ $prm_{sig} \leftarrow \mathsf{SSetup}(1^\kappa)$ Output $prm \leftarrow (prm_{tk}, prm_{sig})$.
$\mathsf{KeyGen_R}(prm)$: Output$(pk_R, sk_R) \leftarrow \mathsf{TKG}(prm_{tk})$.
$\mathsf{KeyGen_S}(prm)$: Output $(pk_S, sk_S) \leftarrow \mathsf{SKG}(prm_{sig})$.
$\mathsf{SC}(prm, pk_R, sk_S, m)$: $\mathsf{tag} \leftarrow pk_S$ $(c_1, K) \leftarrow \mathsf{TEncap}(prm_{tk}, pk_R, \mathsf{tag})$ $\sigma \leftarrow \mathsf{Sign}(prm_{sig}, sk_S, (m\|\|c_1\|\|pk_R))$ $c_2 \leftarrow \mathsf{DEnc}(K, (m\|\|\sigma))$ Output$c \leftarrow (c_1, c_2)$.
$\mathsf{USC}(prm, sk_R, pk_S, c)$: Parse c as (c_1, c_2) $\mathsf{tag} \leftarrow pk_S$ $K/\perp \leftarrow \mathsf{TDecap}(prm_{tk}, sk_R, \mathsf{tag}, c_1)$ (if output is $\perp$, then output $\perp$ and stop.) $(m\|\|\sigma) \leftarrow \mathsf{DDec}(K, c_2)$ If $\mathsf{SVer}(prm_{sig}, pk_S, (m\|\|c_1\|\|pk_R), \sigma) = \perp$ then output $\perp$ and stop. Output m.

図 6.1: TBKEM を用いた構成法 : SC_{tk}

定理 6.1. *TBKEM TK が IND-tag-CCA 安全, かつ DEM D が IND-CCA 安全ならば,* 提案する *Signcryption* 方式 SC_{tk} は *dM-IND-iCCA* の安全性を満たす.

定理 6.2. 署名 S が *sUF-CMA* 安全, かつ *DEM D* が *1* 対 *1* 対応ならば, 提案する *Signcryption* 方式 SC_{tk} は *dM-sUF-iCMA* の安全性を満たす.

定理 6.1 の証明　矛盾を導くために, SC_{tk} の dM-IND-iCCA 安全性を破る攻撃者 $\mathcal{A}$ が存在すると仮定する. ここで以下のゲームを考える (アスタリスク (*) のついた値は Challenge. の際に用いられるとする).

Game_0**:** SC_{tk} に対する通常の dM-IND-iCCA のゲーム.

Game_1**:** このゲームではまず初めに鍵 $K' \in \mathcal{K}$ が一様ランダムに選ばれ, $\mathcal{A}$ のチャレンジ暗号文の要素である c_2^* はこの K' を用いて生成される $(c_2^* \leftarrow \mathsf{DEnc}(K', (m\|\|\sigma)))$. さらに, $(pk_S^*, c_1^*, c_2(\neq c_2^*))$ という形の USC クエリに対しては, c_2 は K' を用いて復号される. 他は Game_0 と同じようにして行う.

Game_i において $\mathcal{A}$ がチャレンジ暗号文を生成する時に使われたビット b を推測する事に成功した場合, そのイベントを Succ_i と定義する. すると,

$$\begin{aligned}\mathsf{Adv}^{\text{dM-IND-iCCA}}_{SC_{tk},\,\mathcal{A}} =& |\Pr[\mathsf{Succ}_0] - \frac{1}{2}| \\ \leq& |\Pr[\mathsf{Succ}_0] - \Pr[\mathsf{Succ}_1]| + |\Pr[\mathsf{Succ}_1] - \frac{1}{2}| \end{aligned} \tag{6.1}$$

が得られる.

証明を完了させるため, 以下の補題を証明する.

補題 6.1. $|\Pr[\mathsf{Succ}_0] - \Pr[\mathsf{Succ}_1]|$ は無視できる値である.

矛盾を導くために, $|\Pr[\mathsf{Succ}_0] - \Pr[\mathsf{Succ}_1]|$ を無視できないと仮定する. 我々は, $\mathcal{A}$ を構成要素として用いて TBKEM TK の IND-tag-CCA を多項式時間で破るシミュレータ $\mathcal{S}$ を構成する. $\mathcal{S}$ は $\mathcal{A}$ に対して以下のように dM-IND-iCCA ゲームのシミュレートを行いつつ利用して, 自身の TK についての IND-tag-CCA ゲームを行う.

Setup. (prm_{tk}, pk_R) を受け取り, $\mathcal{S}$ は $prm_{sig} \leftarrow \mathsf{SSetup}(1^\kappa)$ を実行し, $prm \leftarrow (prm_{tk}, prm_{sig})$ とする. 次に, $\mathcal{S}$ は (prm, pk_R) を入力として $\mathcal{A}$ を起動する.

Phase 1. $\mathcal{S}$ は $\mathcal{A}$ の発行する USC クエリ $(pk_S, (c_1, c_2))$ に対して, 以下のように応答する. まず, $\mathsf{tag} \leftarrow pk_S$ とし, (tag, c_1) を復号クエリとして $\mathcal{CH}$ に問い合わせて K を得る. 次に, 返された値 K から $(m||\sigma) \leftarrow \mathsf{DDec}(K, c_2)$, 及び $\mathsf{SVer}(prm_{sig}, pk_S, (m||c_1||pk_R), \sigma)$ を計算し, SVer の返り値が $\top$ であれば m を $\mathcal{A}$ に返し, そうでなければ $\perp$ を $\mathcal{A}$ に返す.

Challenge. $\mathcal{A}$ が $(m_0, m_1, pk^*_S, sk^*_S)$ を出力したとき, まず $\mathcal{S}$ は pk^*_S をチャレンジタグ $\mathsf{tag}^* = pk^*_S$ として $\mathcal{CH}$ に送信し, (c^*_1, K^*_β) を受け取る. ただし β は IND-tag-CCA ゲームにおける $\mathcal{S}$ のチャレンジビットである. 次に, $\mathcal{S}$ はランダムにコイン $b \in \{0,1\}$ を振り, $\sigma^* \leftarrow \mathsf{Sign}(prm_{sig}, sk^*_S, (m_b||c^*_1||pk_R))$, 及び $c^*_2 \leftarrow \mathsf{DEnc}(K^*_\beta, (m_b||\sigma^*))$ を計算する. 最後に $c^* = (c^*_1, c^*_2)$ を $\mathcal{A}$ のチャレンジ暗号文として $\mathcal{A}$ に返す.

Phase 2. $\mathcal{A}$ の発行する USC クエリ $(pk_S, (c_1, c_2))$ について $\mathcal{S}$ は以下のように応答する.

1. $(pk_S, c_1) = (pk^*_S, c^*_1)$ **である場合**
   $\mathcal{S}$ は $(m||\sigma) \leftarrow \mathsf{DDec}(K^*_\beta, c_2)$, 及び $\mathsf{SVer}(prm_{sig}, pk^*_S, (m||c^*_1||pk_R), \sigma)$ を計算して SVer の返り値が $\top$ であれば m を $\mathcal{A}$ に返し, そうでなければ $\perp$ を $\mathcal{A}$ に返す.
2. **それ以外の場合**
   Phase 1. と同様の動作を行い, 復号結果 $m/\perp$ を返す.

Guess. $\mathcal{A}$ は b' を出力する. $\mathcal{S}$ は $b' = b$ であれば $\beta' = 1$ を自身の推測として出力し, $b' \neq b$ であれば $\beta' = 0$ を出力する.

$\mathcal{S}$ が禁止クエリである $(\mathsf{tag}^*, c_1^*) = (pk_S^*, c_1^*)$ を $\mathcal{CH}$ に発行していない事に注目されたい.

ここで, $\mathcal{S}$ の IND-tag-CCA アドバンテージを計算すると,

$$\begin{aligned}
\mathsf{Adv}^{\text{IND-tag-CCA}}_{TK,\,\mathcal{S}} &= |\Pr[\beta' = \beta] - \frac{1}{2}| \\
&= \frac{1}{2}|1 - \Pr[\beta' = 1|\beta = 1] - \Pr[\beta' = 0|\beta = 0]| \\
&= \frac{1}{2}|\Pr[\beta' = 0|\beta = 1] - \Pr[\beta' = 0|\beta = 0]| \\
&= \frac{1}{2}|\Pr[b' = b|\beta = 1] - \Pr[b' = b|\beta = 0]|
\end{aligned}$$

が得られる.

ここで, $\beta = 1$ の場合, すなわち $K_\beta^* = K_1^*$ が $\mathsf{tag} = pk_S^*$ のもと, c_1^* に対応する本物の共通鍵である場合を考えると, $\mathcal{S}$ が $\mathcal{A}$ に対してチャレンジビットが b であるような Game_0 を完全にシミュレートしている事が分かる. 具体的には, $\mathcal{A}$ の USC クエリに対して完全に Game_0 の場合と同じ様な応答をしており, $\mathcal{A}$ へのチャレンジ暗号文 c^* は Game_0 の場合と同じように生成されている (c_2^* は正しい共通鍵 K_1^* を用いて計算されている). 以上より $b' = b$ が起こる事は, Succ_0 が起こる事を意味する. すなわち, $\Pr[b' = b|\beta = 1] = \Pr[\mathsf{Succ}_0]$ となる.

それに対して $\beta = 0$ の場合, すなわち $K_\beta^* = K_0^*$ が $\{0,1\}^\kappa$ からランダムに選ばれた場合, $\mathcal{A}$ に対して $\mathcal{S}$ は, チャレンジビットが b であるような Game_1 を完全にシミュレートしている. 具体的には, チャレンジ暗号文の要素である c_2^* はランダム鍵 K_0^* を用いて m_b を暗号化して生成し, $(pk_S^*, c_1^*, c_2(\neq c_2^*))$ の形の USC クエリに対しては K_0^* を用いて応答している. 以上より $b' = b$ が起こる事は, Succ_1 が起こる事を意味する. すなわち, $\Pr[b' = b|\beta = 0] = \Pr[\mathsf{Succ}_1]$ となる.

まとめると,

$$\mathsf{Adv}^{\text{IND-tag-CCA}}_{TK,\,\mathcal{S}} = \frac{1}{2}|\Pr[\mathsf{Succ}_1] - \Pr[\mathsf{Succ}_0]|$$

が得られる. これは補題の証明の最初に行った仮定より, 無視できない値となる. ただしこれは TBKEM TK が IND-tag-CCA 安全性を満たすことに矛盾するため, $|\Pr[\mathsf{Succ}_0] - \Pr[\mathsf{Succ}_1]|$ は無視できる値である. 以上より, 補題 6.1 が証明された. □

補題 6.2. $|\Pr[\mathsf{Succ}_1] - 1/2|$ は無視できる値である.

補題 6.2 の証明　矛盾を導くために, $|\Pr[\mathsf{Succ}_1] - 1/2|$ を無視できないと仮定する. 我々は, $\mathcal{A}$ を構成要素として用いて DEM D の IND-CCA を多項式時間で破るシミュレータ $\mathcal{S}$ を構成する. $\mathcal{S}$ は $\mathcal{A}$ に対して以下のように dM-IND-iCCA ゲームのシミュレートを行いつつ利用して, 自身の D についての IND-CCA ゲームを行う.

Setup. $\mathcal{S}$ は $prm_{tk} \leftarrow \mathsf{TSetup}(1^\kappa)$, 及び $prm_{sig} \leftarrow \mathsf{SSetup}(1^\kappa)$ を実行し, $prm \leftarrow (prm_{tk}, prm_{sig})$ とする. 次に, $(pk_R, sk_R) \leftarrow \mathsf{TKG}(prm_{tk})$ を計算し, (prm, pk_R) を入力として $\mathcal{A}$ を起動する.

Phase 1. $\mathcal{S}$ は $\mathcal{A}$ の発行する USC クエリ $(pk_S, (c_1, c_2))$ に対して, 以下のように応答する. まず, $\mathsf{tag} \leftarrow pk_S$ として $K \leftarrow \mathsf{TDecap}(prm_{tk}, sk_R, \mathsf{tag}, c_1)$ を計算する. 次に $(m||\sigma) \leftarrow \mathsf{DDec}(K, c_2)$ を計算し, 最後に $\mathsf{SVer}(prm_{sig}, pk_S, (m||c_1||pk_R), \sigma)$ を計算して, SVer の返り値が $\top$ であれば m を $\mathcal{A}$ に返し, そうでなければ $\perp$ を $\mathcal{A}$ に返す.

Challenge. $\mathcal{A}$ が $(m_0, m_1, pk_S^*, sk_S^*)$ を出力したとき, まず $\mathcal{S}$ は $\mathsf{tag} \leftarrow pk_S^*$ として $(K', c_1^*) \leftarrow \mathsf{TEncap}(pk_R, \mathsf{tag})$ を計算する. 次に, $\sigma_0 \leftarrow \mathsf{Sign}(prm_{sig}, sk_S^*, (m_0||c_1^*||pk_R^*))$, 及び $\sigma_1 \leftarrow \mathsf{Sign}(prm_{sig}, sk_S^*, (m_1||c_1^*||pk_R^*))$ を計算する. さらに $\mathcal{S}$ は, 2 つの平文 $M_0 = (m_0||\sigma_0)$ と $M_1 = (m_1||\sigma_1)$ を $\mathcal{CH}$ に送信し, チャレンジ暗号文 c_2^* を受け取る. 最後に $c^* = (c_1^*, c_2^*)$ を $\mathcal{A}$ に返す.

Phase 2. $\mathcal{A}$ の発行する USC クエリ $(pk_S, (c_1, c_2))$ について $\mathcal{S}$ は以下のように応答する.

1. $(pk_S, c_1) = (pk_S^*, c_1^*)$ **である場合**
   $\mathcal{S}$ は c_2 を USC クエリとして $\mathcal{CH}$ に送信し, $(m||\sigma)$ を得る. 次に $\mathcal{S}$ は, $\mathsf{SVer}(prm_{sig}, pk_S, (m||c_1||pk_R), \sigma)$ を計算して, SVer の返り値が $\top$ であれば m を $\mathcal{A}$ に返し, そうでなければ $\perp$ を $\mathcal{A}$ に返す.
2. **それ以外の場合**
   Phase 1. と同様の動作を行い, 復号結果 $m/\perp$ を返す.

Guess. $\mathcal{A}$ は b' を出力する. $\mathcal{S}$ は b' を自身の推測した値 β' として出力する.

$\mathcal{S}$ が禁止クエリである c_2^* を発行していない事に注目されたい. さらに, $\mathcal{S}$ が $\mathcal{A}$ に対して Game_1 を完全にシミュレートしている事が容易にわかる.

つまり, $\mathcal{S}$ がゲームに勝利する確率を計算すると,

$$\mathsf{Adv}^{\text{IND-CCA}}_{D,\,\mathcal{S}} = |\Pr[\mathsf{Succ}_1] - \frac{1}{2}|$$

が得られる. これはこの補題の証明の最初に行った仮定より, 無視できない値となる. ただしこれは DEM D が IND-CCA 安全性を満たすことに矛盾するため, $|\Pr[\mathsf{Succ}_1] - 1/2|$ は無視できる値である. 以上より, 補題 6.2 が証明された. □

以上をまとめる. 不等式 (6.1), 補題 6.1, 補題 6.2 より, どのような多項式時間アルゴリズム $\mathcal{A}$ においても, $\mathsf{Adv}^{\text{dM-IND-iCCA}}_{SC_{tk},\,\mathcal{A}}$ は無視できる値になる. 以上より定理 6.1 が証明された. □

定理 6.2 の証明　矛盾を導くために, SC_{tk} の dM-sUF-iCMA 安全性を破る攻撃者 $\mathcal{A}$ が存在すると仮定する. 我々は, $\mathcal{A}$ を構成要素として用いて電子署名 S の sUF-CMA を多項式時間で破るシミュレータ $\mathcal{S}$ を構成する. $\mathcal{S}$ は $\mathcal{A}$ に対して以下のように dM-sUF-iCMA ゲームのシミュレートを行いつつ利用して, 自身の S についての sUF-CMA ゲームを行う.

Setup. (prm_{sig}, pk_S) を受け取り，$\mathcal{S}$ は $prm_{tk} \leftarrow \mathsf{TSetup}(1^\kappa)$ を計算し，$prm \leftarrow (prm_{tk}, prm_{sig})$ として，$\mathsf{tag} \leftarrow pk_S$ とする．次に $\mathcal{S}$ は，(prm, pk_S) を入力として $\mathcal{A}$ を起動する．

Query. $\mathcal{S}$ は $\mathcal{A}$ の発行する SC クエリ (pk_R, m) に対して，以下のように応答する．まず，$(c_1, K) \leftarrow \mathsf{TEncap}(prm_{tk}, pk_R, \mathsf{tag})$ を計算する．次に，$\mathcal{CH}$ に対して署名クエリとして $(m||c_1||pk_R)$ を発行し，σ を受け取る．最後に $c_2 \leftarrow \mathsf{DEnc}(K, (m||\sigma))$ を計算し，(c_1, c_2) を $\mathcal{A}$ に返す．

Output. $\mathcal{A}$ が受信者の鍵ペア (pk_R^*, sk_R^*) と暗号文 c^* を出力して終了したら，$\mathcal{S}$ は $((m^*||c_1^*||pk_R^*), \sigma^*)$ を自身の出力として終了する．

$\mathcal{A}$ にとって $\mathcal{S}$ のシミュレーションが完全である事が容易にわかる．具体的には，$\mathcal{A}$ に与えられたパラメータ $prm = (prm_{tk}, prm_{sig})$ と鍵 pk_S は，dM-sUF-iCMA 試行におけるものと完全に同じように分布している．さらに，SC クエリに対する $\mathcal{S}$ の応答も，$\mathcal{S}$ の自身のオラクルへの署名クエリによって完全なものとなっている．

証明を完了させるため，$\mathcal{A}$ が dM-sUF-iCMA 安全性を破るような出力をした場合，常に $\mathcal{S}$ も sUF-CMA 安全性を破る事を示す．まず q を $\mathcal{A}$ の SC クエリの回数とする．また，$i \in \{1, ..., q\}$ において，$(pk_R^{(i)}, m^{(i)})$ を $\mathcal{A}$ の i 番目の SC クエリとし，$(c_1^{(i)}, c_2^{(i)})$ を $\mathcal{A}$ の i 番目のクエリに対する $\mathcal{S}$ の応答とする．さらに，$K^{(i)}$ を，$\mathsf{tag} = pk_S$ として $c_2^{(i)}$ を計算するのに使われた $\mathsf{TEncap}(prm_{tk}, pk_R^{(i)}, \mathsf{tag})$ の出力値である共通鍵とする．

$\mathcal{A}$ が SC_{tk} の dM-sUF-iCMA ゲームで偽造に成功した場合，

$$\begin{aligned}
\mathsf{TDecap}(prm_{tk}, sk_R^*, \mathsf{tag}, c_1^*) &= K^* \neq \perp \\
\mathsf{DDec}(K^*, c_2^*) &= (m^*||\sigma^*) \neq \perp \\
\mathsf{SVer}(prm_{sig}, pk_S, (m^*||c_1^*||pk_R^*), \sigma^*) &= \top \\
{}^\forall i \in \{1, ..., q\} : \qquad (pk_R^*, m^*, c_1^*, c_2^*) &\neq (pk_R^{(i)}, m^{(i)}, c_1^{(i)}, c_2^{(i)})
\end{aligned}$$

となる．ただし $\mathsf{tag} = pk_S$ とする．上記において，　$\mathcal{S}$ が S の sUF-CMA ゲームで偽造に成功するためには，$((m^*||c_1^*||pk_R^*), \sigma^*) = ((m^{(i)}||c_1^{(i)}||pk_R^{(i)}), \sigma^{(i)})$ となるような $i \in \{1, ..., q\}$ が存在してはならない．

ここで，矛盾を導くために，そのような i が存在すると仮定する．このとき，TK の完全性及びこのシミュレーションでは $\mathcal{S}$ が常に同じタグ $\mathsf{tag} = pk_S$ を利用しているという事実から，$pk_R^* = pk_R^{(i)}$ と $c_1^* = c_1^{(i)}$ は $K^* = K^{(i)}$ を意味する．次に，D は 1 対 1 対応であるため，$(K^*, (m^*||\sigma^*)) = (K^{(i)}, (m^{(i)}||\sigma^{(i)}))$ は $c_2^* = c_2^{(i)}$ を意味する．これらを踏まえると，この i では $(pk_R^*, m^*, c_1^*, c_2^*) = (pk_R^{(i)}, m^{(i)}, c_1^{(i)}, c_2^{(i)})$ が成り立つ．しかしこれは上記の $\mathcal{A}$ の勝利条件のうちの 4 番目の式に矛盾するので，そのような i は存在しない．

以上より，$\mathcal{A}$ が偽造に成功すれば $\mathcal{S}$ は必ず sUF-CMA ゲームで偽造に成功するため，$\mathsf{Adv}^{\text{sUF-CMA}}_{S,\,\mathcal{S}} = \mathsf{Adv}^{\text{dM-sUF-iCMA}}_{SC_{tk},\,\mathcal{A}}$ が無視できない値であるという事が得られる．ただしこれは S が sUF-CMA 安全であるという事に矛盾する．

以上より定理 6.2 が証明された．　□

6.2　KEM を用いる一般的構成法

多人数モデルで内部者安全な Signcryption の一般的構成法を KEM などの要素技術を用いて構成する手法を示す.

6.2.1　提案手法

KM : (KSetup, KKG, Encap, Decap) を IND-CCA 安全な KEM とし, D : (DEnc, DDec) を IND-CCA 安全な DEM とする. また, M : (Mac, MVer) を sUF-OT 安全な MAC とする. さらに S : (SSetup, SKG, Sign, SVer) を強偽造不可な署名とする. このとき Signcryption のアルゴリズム SC_{kem} を図 6.2 の様に構成する. ここで KM の鍵空間を $\{0,1\}^{2\kappa}$, D 及び M の鍵空間を $\{0,1\}^{\kappa}$ と仮定する (これは適当な鍵導出関数や擬似ランダム生成器を用いることで常に達成できる).

構成のアイデア　基本的なアイデアは SC_{tk} とあまり変わらない. 要素技術として TBKEM を使わない代わりに, KEM と MAC を用いる. IND-tag-CCA 安全な TBKEM は IND-CCA 安全な KEM と sUF-OT 安全な MAC から構成可能で [4], IND-CCA 安全な DEM は IND-OT 安全な DEM と sUF-OT 安全な MAC から構成可能である [9] とわかっている. 従って, SC_{tk} のように構成した IND-tag-CCA 安全な TBKEM と IND-CCA 安全な DEM を用いる場合, sUF-OT 安全な MAC が共有できる分, 有用である.

6.2.2　スタンダードモデルでの安全性証明

SC_{kem} の安全性は以下の 2 つの定理によって保証されている.

定理 6.3. *KEM KM が IND-CCA 安全, かつ DEM D が IND-OT 安全, MAC M が sUF-OT 安全ならば, 提案する Signcryption 方式 SC_{kem} は dM-IND-iCCA の安全性を満たす.*

定理 6.4. *署名 S が sUF-CMA 安全, かつ DEM D が 1 対 1 対応, MAC M が 1 対 1 対応ならば, 提案する Signcryption 方式 SC_{kem} は dM-sUF-iCMA の安全性を満たす.*

定理 6.3 の証明　矛盾を導くために, SC_{kem} の dM-IND-iCCA 安全性を破る攻撃者 $\mathcal{A}$ が存在すると仮定する. ここで以下のゲームを考える (アスタリスク (*) のついた値は Challenge. の際に用いられるとする).

Game_0**:** SC_{kem} に対する通常の dM-IND-iCCA のゲーム.

Game_1**:** このゲームでは, Phase 2. において, $(pk_S, (c_1^*, c_2, \tau))$ の形をした全ての USC クエリに対して $\perp$ が返される. 他は Game_0 と同じようにして行う.

$\mathsf{Setup}(1^\kappa)$: $prm_{kem} \leftarrow \mathsf{KSetup}(1^\kappa)$ $prm_{sig} \leftarrow \mathsf{SSetup}(1^\kappa)$ Output $prm \leftarrow (prm_{kem}, prm_{sig})$.
$\mathsf{KeyGen_R}(prm)$: Output$(pk_R, sk_R) \leftarrow \mathsf{KKG}(prm_{kem})$.
$\mathsf{KeyGen_S}(prm)$: Output $(pk_S, sk_S) \leftarrow \mathsf{SKG}(prm_{sig})$.
$\mathsf{SC}(prm, pk_R, sk_S, m)$: $(c_1, K) \leftarrow \mathsf{Encap}(prm_{kem}, pk_R)$ $(K_m \| K_a) \leftarrow K$ $\sigma \leftarrow \mathsf{Sign}(prm_{sig}, sk_S, (m \| c_1 \| pk_R))$ $c_2 \leftarrow \mathsf{DEnc}(K_m, (m \| \sigma))$ $\tau \leftarrow \mathsf{Mac}(K_a, (pk_S \| c_1 \| c_2))$ Output$c \leftarrow (c_1, c_2, \tau)$.
$\mathsf{USC}(prm, sk_R, pk_S, c, \tau)$: Parse c as (c_1, c_2, τ) $K \leftarrow \mathsf{Decap}(prm_{kem}, sk_R, c_1)$ $(K_m \| K_a) \leftarrow K$ If $\perp \leftarrow \mathsf{MVer}(K_a, (pk_S \| c_1 \| c_2), \tau)$ then output $\perp$ and stop. $(m \| \sigma) \leftarrow \mathsf{DDec}(K_m, c_2)$ If $\perp \leftarrow \mathsf{SVer}(prm_{sig}, pk_S, \sigma, (m \| c_1 \| pk_R))$ then output $\perp$ and stop. output m

図 6.2: KEM を用いた構成法 : SC_{kem}

Game_2**:** このゲームでは, チャレンジ暗号文が生成されるとき, c_2^*, 及び τ^* を計算するために用いられる共通鍵 (K_m^*, K_a^*) が一様ランダムに選ばれる. 他は Game_1 と同じようにして行う.

Game_i において $\mathcal{A}$ がチャレンジ暗号文を生成する時に使われたビット b を推測する事に成功した場合, そのイベントを Succ_i と定義する. さらに, Game_i において $\mathcal{A}$ が, $\mathsf{MVer}(K_a^*, (pk_S \| c_1^* \| c_2), \tau) = \top$ を満たすような $(pk_S, (c_1^*, c_2, \tau))$ の形の USC クエリを少なくとも一回は発行した場合, そのイベントを Valid_i と定義する. ただし K_a^* は, チャレンジ暗号文の要素である MAC タグ τ^* を生成するのに使われる, MAC のための共通

鍵である. すると,

$$\begin{aligned}\mathsf{Adv}^{\text{dM-IND-iCCA}}_{SC_{kem},\,\mathcal{A}} &= |\Pr[\mathsf{Succ}_0] - \frac{1}{2}| \\ &\leq |\Pr[\mathsf{Succ}_0] - \Pr[\mathsf{Succ}_1]| + |\Pr[\mathsf{Succ}_1] - \Pr[\mathsf{Succ}_2]| + |\Pr[\mathsf{Succ}_2] - \frac{1}{2}| \qquad (6.2)\end{aligned}$$

が得られる.

ここで, Game_0 と Game_1 は, $\mathsf{Valid}_1 = \mathsf{Valid}_0$ が起こるまでは全てにおいて等価である事に注目されたい. つまり,

$$|\Pr[\mathsf{Succ}_0] - \Pr[\mathsf{Succ}_1]| \leq \Pr[\mathsf{Valid}_1] \leq |\Pr[\mathsf{Valid}_1] - \Pr[\mathsf{Valid}_2]| + \Pr[\mathsf{Valid}_2] \qquad (6.3)$$

が得られる.

証明を完了させるため, 以下の補題を証明する.

補題 6.3. $|\Pr[\mathsf{Succ}_1] - \Pr[\mathsf{Succ}_2]|$ は無視できる値である.

補題 6.3 の証明　矛盾を導くために, $|\Pr[\mathsf{Succ}_1] - \Pr[\mathsf{Succ}_2]|$ を無視できないと仮定する. 我々は, $\mathcal{A}$ を構成要素として用いて KEM KM の IND-CCA を多項式時間で破るシミュレータ $\mathcal{S}$ を構成する. $\mathcal{S}$ は $\mathcal{A}$ に対して以下のように dM-IND-iCCA ゲームのシミュレートを行いつつ利用して, 自身の KM についての IND-CCA ゲームを行う.

Setup. (prm_{kem}, pk_R) を受け取り, $\mathcal{S}$ は $prm_{sig} \leftarrow \mathsf{SSetup}(1^\kappa)$ を実行し, $prm \leftarrow (prm_{kem}, prm_{sig})$ とする. 次に, $\mathcal{S}$ は (prm, pk_R) を入力として $\mathcal{A}$ を起動する.

Phase 1. $\mathcal{S}$ は $\mathcal{A}$ の発行する USC クエリ $(pk_S, (c_1, c_2, \tau))$ に対して, 以下のように応答する. まず, c_1 を USC クエリとして $\mathcal{CH}$ に問い合わせて K を得る. 次に, K を $|K_m| = |K_a| = \kappa$ となるように, $(K_m||K_a) = K$ として分割する. さらに, $\mathsf{MVer}(K_a, (pk_S||c_1||c_2), \tau)$ の返り値が $\perp$ であれば $\mathcal{A}$ に $\perp$ を返し, そうでなければ $(m||\sigma) \leftarrow \mathsf{DDec}(K_m, c_2)$, 及び $\mathsf{SVer}(prm_{sig}, pk_S, (m||c_1||pk_R), \sigma)$ を計算する. 最後に, SVer の返り値が $\top$ であれば m を $\mathcal{A}$ に返し, そうでなければ $\perp$ を $\mathcal{A}$ に返す.

Challenge. $\mathcal{A}$ が $(m_0, m_1, pk^*_S, sk^*_S)$ を出力したとき, まず $\mathcal{S}$ はチャレンジ暗号文/鍵対のリクエストを $\mathcal{CH}$ に送信し, (c^*_1, K^*_β) を受け取る. ただし β は IND-CCA ゲームにおける $\mathcal{S}$ のチャレンジビットである. 次に, $\mathcal{S}$ はランダムにコイン $b \in \{0, 1\}$ を振り, K^*_β を $|K^*_m| = |K^*_a| = \kappa$ となるように, $(K^*_m||K^*_a) = K^*_\beta$ として分割する. さらに, $\mathcal{S}$ は $\sigma^* \leftarrow \mathsf{Sign}(prm_{sig}, sk^*_S, (m_b||c^*_1||pk_R))$, $c^*_2 \leftarrow \mathsf{DEnc}(K^*_m, (m_b||\sigma^*))$, 及び $\tau^* \leftarrow \mathsf{Mac}(K^*_a, (pk_S||c^*_1||c^*_2))$ を計算する. 最後に $c^* = (c^*_1, c^*_2, \tau^*)$ を $\mathcal{A}$ のチャレンジ暗号文として $\mathcal{A}$ に返す.

Phase 2. $\mathcal{A}$ の発行する USC クエリ $(pk_S, (c_1, c_2, \tau))$ について, クエリの形が $(pk_S, (c^*_1, c_2, \tau))$ であれば $\mathcal{S}$ は即座に $\perp$ を $\mathcal{A}$ に返すという動作を除いては, Phase 1. と同様の動作を行う.

Guess. $\mathcal{A}$ は b' を出力する. $\mathcal{S}$ は $b' = b$ であれば $\beta' = 1$ を自身の推測として出力し, $b' \neq b$ であれば $\beta' = 0$ を出力する.

$\mathcal{S}$ が禁止クエリである c_1^* を $\mathcal{CH}$ に発行していない事に注目されたい.
ここで, $\mathcal{S}$ の IND-CCA アドバンテージを計算すると,

$$\begin{aligned}\mathsf{Adv}^{\text{IND-CCA}}_{KM,\,\mathcal{S}} &= |\Pr[\beta' = \beta] - \frac{1}{2}| \\ &= \frac{1}{2}|\Pr[\beta' = 0|\beta = 1] - \Pr[\beta' = 0|\beta = 0]| \\ &= \frac{1}{2}|\Pr[b' = b|\beta = 1] - \Pr[b' = b|\beta = 0]|\end{aligned}$$

が得られる.

ここで, $\beta = 1$ の場合, すなわち $K_\beta^* = K_1^* = (K_m^*||K_a^*)$ が c_1^* に対応する本物の共通鍵である場合を考えると, $\mathcal{S}$ が $\mathcal{A}$ に対してチャレンジビットが b であるような Game_1 を完全にシミュレートしている事が分かる. 具体的には, $\mathcal{A}$ の USC クエリに対して完全に Game_0 の場合と同じ様な応答をしており, $\mathcal{A}$ へのチャレンジ暗号文 c^* は Game_0 の場合と同じように生成されている (提案した Signcryption 方式 SC_{kem} の通り, c_2^*, 及び τ^* はそれぞれ正しい共通鍵 K_m^*, 及び K_a^* を用いて計算されている). 以上より $b' = b$ が起こる事は, イベント Succ_1 が起こる事を意味する. すなわち, $\Pr[b' = b|\beta = 1] = \Pr[\mathsf{Succ}_1]$ となる.

それに対して $\beta = 0$ の場合, すなわち $K_\beta^* = K_0^* = (K_m^*||K_a^*)$ が $\{0,1\}^\kappa$ からランダムに選ばれた場合, $\mathcal{A}$ に対して $\mathcal{S}$ は, チャレンジビットが b であるような Game_2 を完全にシミュレートしている. 具体的には, K_0^* がランダムであるため, この場合 K_m, 及び K_a も一様ランダムであり, また, チャレンジ暗号文の要素である c_2^*, 及び τ は共に Game_2 のように正しく計算されている. 以上より $b' = b$ が起こる事は, Succ_2 が起こる事を意味する. すなわち, $\Pr[b' = b|\beta = 0] = \Pr[\mathsf{Succ}_2]$ となる.

まとめると,

$$\mathsf{Adv}^{\text{IND-CCA}}_{KM,\,\mathcal{S}} = \frac{1}{2}|\Pr[\mathsf{Succ}_1] - \Pr[\mathsf{Succ}_2]|$$

が得られる.

これは補題の証明の最初に行った仮定より, 無視できない値となる. ただしこれは KEM KM が IND-CCA 安全性を満たすことに矛盾するため, $|\Pr[\mathsf{Succ}_1] - \Pr[\mathsf{Succ}_2]|$ は無視できる値である. 以上より, 補題 6.3 が証明された. □

補題 6.4. $|\Pr[\mathsf{Valid}_1] - \Pr[\mathsf{Valid}_2]|$ は無視できる値である.

補題 6.4 の証明　補題 6.3 の場合とほぼ同様にして証明できる. まず, 矛盾を導くために, $|\Pr[\mathsf{Valid}_1] - \Pr[\mathsf{Valid}_2]|$ を無視できないと仮定する. 我々は, $\mathcal{A}$ を構成要素として用いて KEM KM の IND-CCA を多項式時間で破るシミュレータ $\mathcal{S}$ を構成する. $\mathcal{S}$ は $\mathcal{A}$ に対して以下のように dM-IND-iCCA ゲームのシミュレートを行いつつ利用して, 自身の KM についての IND-CCA ゲームを行う.

Setup, Phase 1, Challenge, and Phase 2. 補題 6.3 と同様の動作を行う.

Guess. $\mathcal{A}$ は b' を出力して終了する.　次に $\mathcal{S}$ は, $\mathcal{A}$ が $c_1 = c_1^*$, 及び $\mathsf{MVer}(K_a^*, (pk_S||c_1||c_2), \tau) = \top$ を満たす USC クエリ $(pk_S, (c_1, c_2, \tau))$ を発行したかどうかを確認する (ただし K_a^* は, チャレンジ暗号文の要素である MAC タグ τ^* を生成するために用いられる共通鍵である). もしそのようなクエリが見つかれば $\mathcal{S}$ は $\beta' = 1$ を出力し, そうでなければ $\beta' = 0$ を出力して終了する.

ここで, $\mathcal{S}$ の IND-CCA アドバンテージを計算すると,

$$\begin{aligned}\mathsf{Adv}^{\text{IND-CCA}}_{KM,\,\mathcal{S}} &= |\Pr[\beta' = \beta] - \frac{1}{2}| \\ &= \frac{1}{2}|\Pr[\beta' = 1|\beta = 1] - \Pr[\beta' = 1|\beta = 0]|\end{aligned}$$

が得られる.

ここで, $\beta = 1$ の場合, すなわち $K_\beta^* = K_1^* = (K_m^*||K_a^*)$ が c_1^* に対応する本物の共通鍵である場合を考えると, $\mathcal{S}$ が $\mathcal{A}$ に対してチャレンジビットが b であるような Game_1 を完全にシミュレートしている事は確認済みである. このとき, $\mathcal{A}$ が $c_1 = c_1^*$, 及び $\mathsf{MVer}(K_a^*, (pk_S||c_1||c_2), \tau) = \top$ を満たす USC クエリ $(pk_S, (c_1, c_2, \tau))$ を発行したという事は, イベント Valid_1 が起こる事を意味する. つまり, $\Pr[\beta' = 1|\beta = 1] = \Pr[\mathsf{Valid}_1]$ となる.

それに対して, $\beta = 0$ の場合, すなわち $K_\beta^* = K_0^* = (K_m^*||K_a^*)$ が $\{0,1\}^\kappa$ からランダムに選ばれた場合, $\mathcal{A}$ に対して $\mathcal{S}$ は, チャレンジビットが b であるような Game_2 を完全にシミュレートしている. つまり, $\Pr[\beta' = 1|\beta = 0] = \Pr[\mathsf{Valid}_2]$ となる.

まとめると,

$$\mathsf{Adv}^{\text{IND-CCA}}_{KM,\,\mathcal{S}} = \frac{1}{2}|\Pr[\mathsf{Valid}_1] - \Pr[\mathsf{Valid}_2]|$$

が得られる.

これは補題の証明の最初に行った仮定より, 無視できない値となる. ただしこれは KEM KM が IND-CCA 安全性を満たすことに矛盾するため, $|\Pr[\mathsf{Valid}_1] - \Pr[\mathsf{Valid}_2]|$ は無視できる値である. 以上より, 補題 6.4 が証明された. □

補題 6.5. $|\Pr[\mathsf{Succ}_2] - 1/2|$ は無視できる値である.

補題 6.5 の証明　補題 6.5 の証明は DEM D の IND-OT 安全性から明らかなので, 本稿では割愛する. Game_2 においてチャレンジ暗号文の要素である c_2^* のための共通鍵はランダムな値なので, もし $|\Pr[\mathsf{Succ}_2] - 1/2|$ が無視できない値であれば, 我々は $\mathcal{A}$ を用いて D の IND-OT 安全性を破ることができてしまう事に注目されたい.

補題 6.6. $\Pr[\mathsf{Valid}_2]$ は無視できる値である.

補題 6.6 の証明　矛盾を導くために, $\Pr[\mathsf{Valid}_2]$ を無視できないと仮定する. 我々は, $\mathcal{A}$ を構成要素として用いて MAC M の sUF-OT を多項式時間で破るシミュレータ $\mathcal{S}$ を構成する. $\mathcal{S}$ は $\mathcal{A}$ に対して以下のように dM-IND-iCCA ゲームのシミュレートを行いつつ利用して, 自身の M についての sUF-OT ゲームを行う.

Setup. まず $\mathcal{S}$ は $prm_{kem} \leftarrow \mathsf{KSetup}(1^\kappa)$, 及び $prm_{sig} \leftarrow \mathsf{SSetup}(1^\kappa)$ を実行し, $prm \leftarrow (prm_{kem}, prm_{sig})$ とする. 次に, $\mathcal{S}$ は $(pk_R, sk_R) \leftarrow \mathsf{KKG}(prm_{kem})$ を実行し, (prm, pk_R) を入力として $\mathcal{A}$ を起動する.

Phase 1. $\mathcal{S}$ は受信者の秘密鍵を所持しているので, Game_2 のように $\mathcal{A}$ の発行する全ての USC クエリ $(pk_S, (c_1, c_2, \tau))$ に対して応答できる.

Challenge. $\mathcal{A}$ が $(m_0, m_1, pk_S^*, sk_S^*)$ を出力したとき, $\mathcal{S}$ は以下のように応答する. まず $(K', c_1^*) \leftarrow \mathsf{Encap}(prm_{kem}, pk_R)$ を計算する. 次に, ランダムにコイン $b \in \{0,1\}$ を振り, $K_m^* \in \{0,1\}^\kappa$ を一様ランダムにを選び, $\sigma^* \leftarrow \mathsf{Sign}(prm_{sig}, sk_S^*, (m_b||c_1^*||pk_R^*))$, 及び $c_2^* \leftarrow \mathsf{DDec}(K_m^*, (m_b||\sigma^*))$ を計算する. さらに, MAC クエリ $(pk_S^*||c_1^*||c_2^*)$ を発行し, $\mathcal{CH}$ から MAC タグ τ^* を受け取る. 最後に, $c^* = (c_1^*, c_2^*, \tau^*)$ を $\mathcal{A}$ のチャレンジ暗号文として $\mathcal{A}$ に返す.

Phase 2. $\mathcal{S}$ は sk_R を用いて, Game_2 のように $\mathcal{A}$ の発行する全ての USC クエリに対して応答する.

Guess. $\mathcal{A}$ は b' を出力して終了したら, $\mathcal{S}$ は $\mathcal{A}$ の USC クエリの中から一様ランダムに USC クエリ $(pk_S, (c_1, c_2, \tau))$ を 1 つ選び, $((pk_S||c_1||c_2), \tau)$ を自身の出力として終了する.

$\mathcal{A}$ にとって $\mathcal{S}$ のシミュレーションが完全である事が容易にわかる. 証明を完了するため, まず q を $\mathcal{A}$ の USC クエリの回数とする. もしイベント Valid_2 が起こったならば, $\mathcal{A}$ は $c_1 = c_1^*$, 及び $\mathsf{MVer}(K_a^*, (pk_S||c_1^*||c_2), \tau) = \top$ を満たす USC クエリ $(pk_S, (c_1, c_2, \tau))$ を少なくとも一回は発行した事になる. さらに, $\mathcal{A}$ は SC_{kem} の dM-IND-iCCA 安全性を破る攻撃者なので, $(pk_S, (c_1, c_2, \tau)) \neq (pk_S, (c_1, c_2, \tau))$ が得られる. $\mathcal{S}$ は一様ランダムに USC クエリを 1 つ選ぶため, イベント Valid_2 の定義より, $\mathcal{S}$ が偽造に成功する (sUF-OT ゲームに勝利する) 確率は少なくとも $1/q$ 以上である.

以上より, $\mathcal{S}$ の sUF-OT アドバンテージを計算すると,

$$\mathsf{Adv}^{\text{sUF-OT}}_{M,\,\mathcal{S}} \geq \frac{1}{q} \Pr[\mathsf{Valid}_2]$$

が得られる. これはこの補題の証明の最初に行った仮定より, 無視できない値となる. ただしこれは MAC M が sUF-OT 安全性を満たすことに矛盾するため, $\Pr[\mathsf{Valid}_2]$ は無視できる値である. 以上より, 補題 6.6 が証明された. □

以上をまとめる. 不等式 (6.2), (6.3), 補題 6.3〜6.6 より, どのような多項式時間アルゴリズム $\mathcal{A}$ においても, $\mathsf{Adv}^{\text{dM-IND-iCCA}}_{SC_{kem},\,\mathcal{A}}$ は無視できる値になる. 以上より定理 6.3 が証明された. □

定理 6.4 の証明　矛盾を導くために, SC_{kem} の dM-sUF-iCMA 安全性を破る攻撃者 $\mathcal{A}$ が存在すると仮定する. 我々は, $\mathcal{A}$ を構成要素として用いて電子署名 S の sUF-CMA を多項式時間で破るシミュレータ $\mathcal{S}$ を構成する. $\mathcal{S}$ は $\mathcal{A}$ に対して以下のように dM-sUF-iCMA ゲームのシミュレートを行いつつ利用して, 自身の S についての sUF-CMA ゲームを行う.

Setup. (prm_{sig}, pk_S) を受け取り, $\mathcal{S}$ は $prm_{kem} \leftarrow \mathsf{KSetup}(1^\kappa)$ を計算し, $prm \leftarrow (prm_{kem}, prm_{sig})$ とする. 次に $\mathcal{S}$ は, (prm, pk_S) を入力として $\mathcal{A}$ を起動する.

Query. $\mathcal{S}$ は $\mathcal{A}$ の発行する SC クエリ (pk_R, m) に対して, 以下のように応答する. まず, $(c_1, K) \leftarrow \mathsf{Encap}(prm_{kem}, pk_R)$ を計算する. 次に, $\mathcal{CH}$ に対して署名クエリとして $(m||c_1||pk_R)$ を発行し, σ を受け取る. さらに, $(K_m||K_a) \leftarrow K$ として $c_2 \leftarrow \mathsf{DEnc}(K_m, (m||\sigma))$ を計算する. 最後に $\tau \leftarrow \mathsf{Mac}(K_a, (pk_S||c_1||c_2))$ を計算し, (c_1, c_2, τ) を $\mathcal{A}$ に返す.

Output. $\mathcal{A}$ が受信者の鍵ペア (pk_R^*, sk_R^*) と暗号文 c^* を出力して終了したら, $\mathcal{S}$ は $((m^*||c_1^*||pk_R^*), \sigma^*)$ を自身の出力として終了する.

$\mathcal{A}$ にとって $\mathcal{S}$ のシミュレーションが完全である事が容易にわかる. 具体的には, $\mathcal{A}$ に与えられたパラメータ $prm = (prm_{kem}, prm_{sig})$ と鍵 pk_S は, dM-sUF-iCMA ゲームにおけるものと完全に同じように分布している. さらに, SC クエリに対する $\mathcal{S}$ の応答も, $\mathcal{S}$ の自身のオラクルへの署名クエリによって完全なものとなっている.

証明を完了させるため, $\mathcal{A}$ が dM-sUF-iCMA 安全性を破るような出力をした場合, 常に $\mathcal{S}$ も sUF-CMA 安全性を破る事を示す. まず q を $\mathcal{A}$ の SC クエリの回数とする. また, $i \in \{1, ..., q\}$ において, $(pk_R^{(i)}, m^{(i)})$ を $\mathcal{A}$ の i 番目の SC クエリとし, $(c_1^{(i)}, c_2^{(i)}, \tau^{(i)})$ を $\mathcal{A}$ の i 番目のクエリに対する $\mathcal{S}$ の応答とする. さらに, $K^{(i)} = (K_m^{(i)}, K_a^{(i)})$ を, $c_2^{(i)}$ 及び $\tau^{(i)}$ を計算するのに使われた $\mathsf{Encap}(prm_{kem}, pk_R^{(i)})$ の出力値である共通鍵とする.

$\mathcal{A}$ が SC_{kem} の dM-sUF-iCMA ゲームで偽造に成功した場合,

$$\begin{aligned}
\mathsf{Decap}(prm_{kem}, sk_R^*, c_1^*) &= K^* = (K_m^*||K_a^*) \neq \perp \\
\mathsf{MVer}(K_a^*, (pk_S||c_1^*||c_2^*), \tau^*) &= \top \\
\mathsf{DDec}(K_m^*, c_2^*) &= (m^*||\sigma^*) \neq \perp \\
\mathsf{SVer}(prm_{sig}, pk_S, (m^*||c_1^*||pk_R^*), \sigma^*) &= \top \\
{}^\forall i \in \{1, ..., q\} : \qquad (pk_R^*, m^*, c_1^*, c_2^*, \tau^*) &\neq (pk_R^{(i)}, m^{(i)}, c_1^{(i)}, c_2^{(i)}, \tau^{(i)})
\end{aligned}$$

となる. 上記において, $\mathcal{S}$ が S の sUF-CMA ゲームで偽造に成功するためには, $((m^*||c_1^*||pk_R^*), \sigma^*) = ((m^{(i)}||c_1^{(i)}||pk_R^{(i)}), \sigma^{(i)})$ となるような $i \in \{1, ..., q\}$ が存在してはならない.

ここで, 矛盾を導くために, そのような i が存在すると仮定する. このとき, KM の完全性及びこのシミュレーションでは $\mathcal{S}$ が常に $sk_R^* = sk_R^{(i)}$ を使用しているという事実から, $c_1^* = c_1^{(i)}$ は $K^* = K^{(i)}$, すなわち $(K_m^*, K_a^*) = (K_m^{(i)}, K_a^{(i)})$ を意味する. 次に, D は 1 対 1 対応であるため, $(K_m^*, (m^*||\sigma^*)) = (K_m^{(i)}, (m^{(i)}||\sigma^{(i)}))$ は $c_2^* = c_2^{(i)}$ を意味する. さらに, M

は1対1対応であるため, $(K_a^*, (m^*||c_1^*||pk_R^*)) = (K_a^{(i)}, (m^{(i)}||c_1^{(i)}||pk_R^{(i)}))$ は $\tau^* = \tau^{(i)}$ を意味する. これらを踏まえると, この i では $(pk_R^*, m^*, c_1^*, c_2^*, \tau^*) = (pk_R^{(i)}, m^{(i)}, c_1^{(i)}, c_2^{(i)}, \tau^{(i)})$ が成り立つ. しかしこれは上記の $\mathcal{A}$ の勝利条件のうちの5番目の式に矛盾するので, そのような i は存在しない.

以上より, $\mathcal{A}$ が偽造に成功すれば $\mathcal{S}$ は必ず sUF-CMA ゲームで偽造に成功するため, $\mathsf{Adv}^{\text{sUF-CMA}}_{S,\,\mathcal{A}} = \mathsf{Adv}^{\text{dM-sUF-iCMA}}_{SC_{kem},\,\mathcal{S}}$ が無視できない値であるという事が得られる. ただしこれは S が sUF-CMA 安全であるという事に矛盾する.

以上より定理 6.4 が証明された. □

Chapter 7　議論

本章では, 5 章での提案方式に関するより進んだ議論, 及び 6 章での提案手法に関するより進んだ議論を行う.

7.1 提案する TBKEM の強化手法を用いた Signcryption への応用

本節は, 提案した TBKEM の安全性の強化手法がスタンダードモデルで内部者安全性を証明可能で, かつ効率的な Signcryption を構成するのに有用なことを示す. 具体的には, [36] における TBKEM と署名のうち, 4.2 節で詳細に述べた Signcryption 結合可能性と呼ばれる性質を満たす方式, 及び DEM を用いた Signcryption の構成法において必要とされる TBKEM の方の Signcryption 結合可能性について, もし提案方式の変換前の IND-stag-CCA 安全な TBKEM がこの性質をある署名方式と満たした場合, 提案方式でのカメレオンハッシュを用い変換後の IND-tag-CCA 安全な TBKEM も満たすことを示す.

まず, 分割可能な TBKEM(暗号化アルゴリズムが TE_1 と TE_2, 乱数空間が $\mathcal{R}$) をカメレオンハッシュ(ハッシュ鍵 hk, 乱数空間 $\mathcal{R}_{cmh}$) と本提案手法で組み合わせると, 変換後の TBKEM の暗号化アルゴリズムの乱数空間は $\mathcal{R}' = \mathcal{R} \times \mathcal{R}_{cmh}$ となり, その分割されたアルゴリズム TE'_1 と TE'_2 はそれぞれ

$\mathsf{TE}'_1(prm, R)$ (ただし $R = (r, r_{cmh}) \in \mathcal{R}'$):
　$c_1 \leftarrow \mathsf{TE}_1(prm, r)$;
　Output $c'_1 \leftarrow (c_1, r_{cmh})$.
$\mathsf{TE}'_2(prm, pk', \mathsf{tag}, R)$ (ただし $pk' = (pk, hk)$ 及び $R = (r, r_{cmh}) \in \mathcal{R}'$):
　$\mathsf{tag}' \leftarrow \mathsf{CMH}(hk, \mathsf{tag}; r_{cmh})$;
　Output $(c'_2, K) \leftarrow \mathsf{TE}_2(prm, pk, \mathsf{tag}', r)$

となる. 上記はもちろん分割可能な性質のうち (ii)(1) を満たしている (4.2 節を参照). また, prm, $pk' = (pk, hk)$, tag, $c'_1 = (c_1, r_{cmh})$ が全て固定されたとする. すると, カメレオンハッシュへの入力が全て固定されるため, tag' も固定される. prm, pk, tag', 及び c_1 が固定されるため, 性質 (ii)(2) より, $c = (c_1, c_2)$ としたときに $\mathsf{TDecap}(prm, sk, \mathsf{tag}', c) = K$ となる (c'_2, K) は高々一組しかない. 従って, prm, $pk' = (pk, hk)$, tag, $c'_1 = (c_1, r_{cmh})$ が全て固定されたとき, $c' = (c'_1, c'_2)$ としたときに $\mathsf{TDecap}'(prm, sk, \mathsf{tag}, c') = K$ となる (c'_2, K) もやはり一つしかない. 以上より, 提案した変換手法は, 分割可能の要件を保存する.

5.3 節で示した自明な手法強化された IND-tag-CCA 安全な TBKEM では, そもそも分割可能性を満たせない. TBKEM の共通鍵として用いるために暗号化される K は秘匿性を必要とする情報であり, (ii) を満たす二つの決定的アルゴリズムが構成できないためである.

また, 2.2.2 節で紹介した Boyen ら [16] による KEM や Hanaoka と Kurosawa [25] による KEM は, IND-stag-CCA 安全な TBKEM に実質的にコスト無しで変換可能であり, しかも Waters 署名 [44], Camenish-Lyshanskaya(CL) 署名 [17], Boneh-Shen-Waters(BSW) 署名 [14] 方式, 及び CL 署名に BSW 署名 [14] での技術を適用した方式 (CL' 署名と呼ぶ) と Signcryption 結合可能である.

また, 提案後の変換方式は, 提案した変換の変換前後で秘密鍵の要素は増えないため, 変換前の方式が Signcryption 結合可能であれば, (iii) の条件も満たす.

従って, もし [36] で挙げられている Signcryption 結合可能でかつ IND-stag-CCA 安全な TBKEM 方式で, 具体的なカメレオンハッシュ(2.6.1 節で示した構成など) と組み合わせた後でも Signcryption 結合可能性の条件 (i) を満足する署名方式があれば, [36] の結果に最も強いモデルで内部者安全な Signcryption 方式が構成可能である.

幸いなことに, [36] で挙げられている署名方式のうち, BSW 署名 [14] と CL' 署名 [17, 14] は [36] で挙げられている全ての IND-stag-CCA 安全な TBKEM と Signcrpytion 結合可能性の条件の (i) を満たす. 直感的には, これらの方式もカメレオンハッシュを署名の構成要素としており, カメレオンハッシュの乱数は署名の一部として送られているからである. 詳細は原論文を参照されたい.

7.2　提案するSigncryptionの一般的構成法に関する議論

本節では6章での提案手法の効率評価, 否認防止に関する議論, 及び新しく得られる具体的なSigncryptionの構成について述べる.

7.2.1　効率評価

表7.1において, 具体的なスタンダードモデルでのSigncryptionの比較を行う. この図の中で, tBMW1は[36, Sect.7.2]の表記法に基づき, Boyenら[15]による公開鍵方式から得られる(TBEと)TBKEMによる構成を表しており, [1] BMW2はBoyenら[16, Sect.4]によるKEMによる構成を表している. MMS-StTE(X, Y)は[36, Sect.5]の "Sign-then-Tag-based-Encrypt" を表しており, TBE Xと電子署名Yが要素技術として使われている方式を表している. MMS-SC(X, Y)はSigncryption結合可能性[36, Sect.6]を満たすTBKEM Xと電子署名Yから構成されるSigncryptionを表している. SC_{tk}(X, Y) (同様に, SC_{kem}(X, Y))は6章で述べた, TBKEM(同様に, KEM)Xと電子署名Yを構成要素としたSigncryptionを表している.

表7.1からわかるように, ランダムオラクルなしにdM-IND-iCCAとdM-sUF-iCMAを同時に達成する手法は我々の構成法のみである. SC_{tk}(tBMW1, BB)はTanやMMS-StTE(tBMW1, BB)と比較した場合全てにおいて効率が良い. また, SC_{tk}(tBMW1, BB), MMS-SC(tBMW1, Waters), 及びMMS-SC(tBMW1, BSW)において達成された安全性/仮定, 計算コストの間でトレードオフがあるとわかる. しかし, q-SDH仮定を許して内部者に対する強偽造不可能性が必要であるならば, SC_{tk}(tBMW1, BB)は最も良い選択であるということを強調したい. また, SC_{kem}(BMW2, BB)とTanはともにユーザー(送信者と受信者)公開鍵サイズが一定であり, SC_{kem}(BMW2, BB)はTanよりも全てにおいて効率がよい.

表においては比較のために具体的な要素技術としてtBMW1, BMW2, 及びBBを選んでいるが, 提案手法であるSC_{tk}とSC_{kem}は一般的構成法である(これは[36]の構成法でも同じである)ので, 他にも数多くの具体的な構成が可能である. 例えば, 離散対数系の仮定ではなく素因数分解系の仮定を用いて内部者安全なSigncryptionを構成したいのであれば, 素因数分解の困難さによる仮定から安全性が証明されているHofheinzとKiltz[26]によるKEMを利用でき(そのTBKEM版は[36]に見られるものから得られる), RSA仮定から安全性が証明されているHohenbergerとWaters[27]による電子署名を利用する事ができる. 他にも, [5]による手法のようにIDベース暗号/署名に関する最近の進歩から, 格子問題系の仮定から内部者安全なSigncryptionを構成できる([13]によるとIDベース暗号から常にCCA安全なKEMを構成する事ができる). また, 5章において提案したTBKEMの強化手法をこの構成法に応用できる. すなわち, IND-stag-CCA安全性を持つTBKEMを強化手法によりIND-tag-CCA安全なTBKEMに変換し, 本構成法の要素技術として利用することも可能である.

[1] Boyenらの公開鍵暗号(及びKEM) [16]では構成要素として(ターゲット)衝突困難ハッシュ関数が使われており, 本来の入力に連結してタグを入力すれば安全なTBE(TBKEM)となる事が[36]で指摘されている. 後述のHofheinzとKiltzのKEM [26]でも同様にしてTBKEMへと変換可能.

表 7.1: 多人数モデルにおける内部者安全な Signcryption の既存手法と提案手法の比較

Scheme	秘匿性/ 仮定	偽造不可能性/ 仮定	計算コスト SC / USC	暗号文オーバーヘッド / ビット
Tan [43]	dM-IND-iCCA/ DBDH	dM-sUF-iCMA(KR)/ q-SDH	[3,2;0]/ [3,1;4]	$3\|\mathbb{G}_p\| + 2\|\mathbb{Z}_p\|$/ 800
MMS-StTE (tBMW1,BB [12])	dM-IND-iCCA/ DBDH	dM-wUF-iCMA/ q-SDH	[4,0;0] + 1W/ [1,1;2]	$3\|\mathbb{G}_p\| + \|\mathbb{Z}_p\|$/ 640
MMS-SC (tBMW1,Waters [44])	dM-IND-iCCA/ DBDH	dM-wUF-iCMA(KR)/ co-CDH	[4,0;0]/ [1,0;3] + 1W	$3\|\mathbb{G}_p\|$/ 480
MMS-SC (tBMW1,BSW [14])	dM-IND-iCCA/ DBDH	dM-sUF-iCMA(KR)/ co-CDH	[4,1;0]/ [1,1;3] + 1W	$3\|\mathbb{G}_p\| + \|\mathbb{Z}_p\|$/ 640
Ours: SC_{tk} (tBMW1,BB [12])	dM-IND-iCCA/ DBDH	dM-sUF-iCMA/ q-SDH	[4,0;0] + 1W/ [1,1;2]	$3\|\mathbb{G}_p\| + \|\mathbb{Z}_p\|$/ 640
Ours: SC_{kem} (BMW2,BB [12])	dM-IND-iCCA/ DBDH	dM-sUF-iCMA/ q-SDH	[3,1;0]/ [1,1;2]	$3\|\mathbb{G}_p\| + \|\mathbb{Z}_p\| + \|MAC\|$/ 720

“秘匿性” と “偽造不可能性” の列は安全性の仮定および達成される安全性を表記している. この中で (KR) のついた安全性は鍵登録 [36] の必要性を意味する. “計算コスト” の列は暗号化 (SC) と復号 (USC) における計算コストを表記している. ここで $[a, b; c]$ は, それぞれ a が指数計算, b が多重指数計算, c がペアリング演算を表し, W はいわゆる Waters ハッシュ [44] を表す (その他の掛け算, ハッシュ関数, 共通鍵暗号の計算は無視する). “暗号文オーバーヘッド” の列において, “暗号文オーバーヘッド” は暗号文サイズと平文サイズの差を意味する. $|\mathbb{G}|$ は双線形ペアリングにおける楕円曲線の群の要素の大きさを表し, $|\mathbb{Z}_p|$ は指数の大きさを表し, |MAC| は (sUF-OT 安全な)MAC タグの大きさを表す. 同列には 80 ビットの安全性を想定した場合のサイズの数値例も示した. この場合, $|\mathbb{G}| = |\mathbb{Z}_p| = 160$ で $|\text{MAC}| = |IV| = 80$ とした. DEM は暗号文オーバーヘッドが 0 であると仮定 [38] し, IND-OT 安全か IND-CCA 安全かは考慮しない.

また, この表では 4.1 節で述べたような比較的初期に提案された Signcryption とは比較を行っていない. この表による比較の目的は, Signcryption の安全性が明確に定義された上で, 多人数モデルで内部者安全という最も強い安全性を達成する構成との比較を行うことにある. 従って, ランダムオラクルモデルを利用する構成とも比較を行っておらず, 比較対象の構成法は全てスタンダードモデルによるものである. 実際には, ランダムオラクルモデルを用いると非常に効率のよい構成が可能となるが, 現実に存在し得ないランダムオラクルを仮定する事で安全性における問題点を持ってしまうため, この表では比較対象から外している.

7.2.2　否認防止について

一般的な電子署名に求められる安全性要件として, “否認防止” がある. これは, 署名者はあるメッセージに一旦署名を生成すると, その署名を作成した事実を後で否認できない, という要件である. 電子署名の場合と同様に, Signcryption の場合も否認防止が達成されている事が必要である. 本稿ではこれまでデータの完全性として偽造不可能性についての言及を行ってきたが, 本節において提案した Signcryption 方式が, 否認防止もまた達成している事について述べる. また, 偽造不可能性の場合とは異なり, 否認防止については数学的な安全性定義はあまり行われない.

ここで, 送信者が本稿の提案手法によって構成された Signcryption を利用して暗号文 c を作成して受信者に送信し, 後になって送信者が署名を作成した事実を否認した場合を考える. これに対して受信者は裁判官などの第三者に検証してもらうというプロセスをとる. この際, 以下の二つの場合を考える.

1. **送信者が m というメッセージに署名をして送信したかどうかという点のみを検証する必要がある場合**
   この場合, 受信者は復号結果である m と σ を第三者に公開し, 第三者は $\mathsf{SVer}(prm, vk, m, \sigma)$ を実行する事によって検証を行う. つまり, この検証によって署名が有効であった場合, 受信者が悪意を持って偽の m と σ を発行していたという事は有り得ないため, 送信者が署名を作成した事実が判明する (署名鍵を知らない受信者はメッセージとそれに対する正しい署名のペアを出力できない事が sUF-CMA 安全性の定義であるため).

2. **送信者が作成して受信者に送信した暗号文 c が, 受信者によって正しく復号されていて, かつその復号されたメッセージと署名の検証をする必要がある場合**
   この場合, 受信者は第三者に対して, 暗号文 $c = (c_1, c_2)$, メッセージ m, 署名 σ, 共通鍵 K を公開する. 第三者は受信者の秘密鍵 sk_R までは知る事ができないため, m が c に対する正しい復号結果であるという事を確認するには, 受信者との間でゼロ知識証明を用いる必要がある. それ以外は 1. の場合と同様である.

Signcryption は, 受信者の秘密鍵を持たない第三者は暗号文 c を復号できないため, 上記の 2. のような場合において (ゼロ知識証明などの他のプロセス無しでは) 署名の検証を行う事ができない. しかし実際には, そのような場合は滅多に起こらない (1. の場合

は起こり得ても 2. の場合はあまり起こらない) と仮定して, ゼロ知識証明という重いプロセスを利用する事で検証を実現している.

7.2.3　新しく得られる Signcryption 方式の具体例

提案する Signcryption の一般的構成法に具体的な要素技術を適用させた方式のアルゴリズムを紹介する.

[36] で提案されている IND-tag-CCA 安全な TBKEM のうち, Boyen らによる公開鍵方式 [16] を基にして得られる TBKEM と, Boneh と Boyen による sUF-CMA 安全な電子署名 [12] を要素技術として用いた方式 (表 7.1 中の SC_{tk}(tBMW1, BB)) を, 図 7.1 に示す.

表 7.1 からわかるように, この方式は, [36] で提案された一般的構成法に, Boyen らによる公開鍵方式 [16] から得られる IND-tag-CCA 安全な TBKEM と, Boneh と Boyen による sUF-CMA 安全な電子署名 [12] を組み合わせた, 現在スタンダードモデルで最も効率がよい Signcrption 方式 (表 7.1 中の MMS-StTE(tBMW1, BB)) と全く同等の安全性と暗号文サイズを持つ. また, 安全性については, 提案手法による構成の方がより強い偽造不可能性を達成するという事に注目されたい.

$\mathsf{Setup}(1^\kappa)$:
Pick bilinear groups $(\mathbb{G}, \hat{\mathbb{G}}, \mathbb{G}_T)$ (order p)
with $e : \mathbb{G} \times \hat{\mathbb{G}} \to \mathbb{G}_T$ and $\psi : \hat{\mathbb{G}} \to \mathbb{G}$
$\hat{g} \leftarrow \hat{\mathbb{G}}, \quad g \leftarrow \psi(\hat{g}), \quad \mathtt{Z} \leftarrow e(g, \hat{g})$
Pick a CRHF $H : \{0,1\}^* \leftarrow \{0,1\}^n$ s.t. $\{0,1\}^n \subset \mathbb{Z}_p$
Pick a DEM $(\mathsf{DEnc}, \mathsf{DDec})$ with key space $\mathbb{G}_T$.
Output $prm \leftarrow (p, \mathbb{G}, \hat{\mathbb{G}}, e, \psi, g, \hat{g}, H, \mathsf{DEM})$.

$\mathsf{KeyGen_R}(prm)$:
$u', u_1, \ldots, u_n \leftarrow \mathbb{Z}_p, \quad U' \leftarrow g^{u'}, \quad U_i \leftarrow g^{u_i}$
$\alpha \leftarrow \mathbb{Z}_p, \quad \hat{h} \leftarrow \hat{g}^\alpha, \quad \mathtt{Z}_R \leftarrow e(g, \hat{g})^\alpha \in \mathbb{G}_T$
$pk_R \leftarrow (\mathtt{Z}_R, U', U_1, \ldots, U_n), \quad sk_R \leftarrow (\hat{h}, u', u_1, \ldots, u_n)$
Output (pk_R, sk_R).

$\mathsf{KeyGen_S}(prm)$:
$x, y \leftarrow \mathbb{Z}_p^*, \quad u \leftarrow \hat{g}^x, \quad v \leftarrow \hat{g}^y, \quad \mathtt{Z}_S \leftarrow e(g, \hat{g}) \in \mathbb{G}_T$
$pk_S \leftarrow (\mathtt{Z}_S, u, v), \quad sk_S \leftarrow (x, y)$
Output (pk_S, sk_S).

$\mathsf{SC}(prm, pk_R, sk_S, m)$:
Parse pk_R as $(\mathtt{Z}_R, U', U_1, \ldots, U_n)$ and sk_S as (x, y).
$r, s \leftarrow \mathbb{Z}_p^*, \quad c_1 \leftarrow g^r, \quad \mathsf{tag} \leftarrow pk_S, \quad t \leftarrow H(\mathsf{tag}||c_1)$
Let each t_i be i-th bit of t.
$c_2 \leftarrow (U' \prod_{i=1}^{n} U_i^{t_i})^r, \quad \mathtt{K} \leftarrow \mathtt{Z}_R^r$
$\sigma \leftarrow g^{\frac{1}{(m||c_1||pk_R)+x+ys}}, \quad c_3 \leftarrow \mathsf{DEnc}(\mathtt{K}, (m||\sigma))$
Output $c \leftarrow (c_1, c_2, c_3)$.

$\mathsf{USC}(prm, sk_R, pk_S, c)$:
Parse pk_S as $(\mathtt{Z}_S, u, v)$ and sk_R as $(\hat{h}, u', u_1, \ldots, u_n)$.
Parse c as (c_1, c_2, c_3). $\quad \mathsf{tag} \leftarrow pk_S, \quad t \leftarrow H(\mathsf{tag}||c_1)$
Let each t_i be i-th bit of t.
If $c_2 \neq c_1^{u' + \Sigma_{i=1}^n u_i t_i}$ then output $\perp$ and stop.
$\mathtt{K} \leftarrow e(c_1, \hat{h}), \quad (m||\sigma) \leftarrow \mathsf{DDec}(\mathtt{K}, c_3)$
If $e(\sigma, u \cdot \hat{g}^m \cdot v^s) \neq \mathtt{Z}_S$ then output $\perp$ and stop.
Output m.

図 7.1: 新しく得られる Signcryption 方式の具体例.　e は双線形写像であり, 本稿では $\hat{\mathbb{G}}$ から $\mathbb{G}$ へと同型写像 ψ がある双線形群を考える. 図ではハット付き文字は $\hat{\mathbb{G}}$ の要素を, truetype フォントの文字は $\mathbb{G}_T$ の要素を表している.

Chapter 8　結論

本研究では, 公開鍵暗号と電子署名を組み合わせた機能を持つ, Signcryption について着目し, その最も強い安全性として定義されている “多人数モデルで内部者に対する秘匿性と偽造不可能性” をスタンダードモデルであらゆる制限なしに満たす一般的構成法を, はじめて提案した.

提案した一般的構成法は 2 つあり, 第一の方式では IND-tag-CCA 安全な TBKEM, 1 対 1 対応の性質を持つ IND-CCA 安全な DEM, 及び sUF-CMA 安全な署名を用いて構成した. また, 第二の方式では IND-CCA 安全な KEM, 1 対 1 対応の性質を持つ IND-OT 安全な DEM, 1 対 1 対応の性質をもつ一回安全な MAC, 及び sUF-CMA 安全な署名を用いて構成した.

効率の面においても, これまで提案された Signcryption 方式の中で安全性が強く最も効率の良い [36] による構成法とほぼ同程度の効率を達成する事ができた. また, [36] と同様に, 上記手法の利点は提案手法から効率的な Signcryption の構成ができるだけでなく, “一般的構成法”, すなわち既に確立された構成要素の安全性の結果を利用できるという点にもあると強調したい. そのため, 本稿で提案した 2 つの構成法は将来的に具体的に構成されるであろう具体的な Signcryption 方式の良いベンチマークになる事を信じている.

また, 本研究のもう一つの成果として, TBKEM の強化手法を提案した. 具体的には, 選択的タグ CCA 安全な TBKEM から適応的タグ CCA 安全な TBKEM への効率的な変換をするため, カメレオンハッシュと呼ばれる特殊なハッシュ関数を用いた. 選択的タグ CCA 安全な TBKEM から適応的タグ CCA 安全な TBKEM への変換については, 同目的を達成する自明な手法が存在するが, 本手法は自明な手法と比べ変換後の暗号文サイズを小さくできるため, 変換後の暗号文サイズの増加を抑えたい場合には本提案手法は便利である. さらに, この強化手法の応用先として, 前述の Signcryption の一般的構成法の要素技術が考えられる. 本手法を Signcryption の一般的構成法に適用させることで, 選択的タグ安全な TBKEM だけでなく適応的タグ安全な TBKEM も Signcryption の要素技術として利用できるため, Signcryption の具体的な構成の幅が広がったと言えよう.

謝辞

本論文の作成にあたり, 2 年間を通して常にご指導をして頂きました東京大学生産技術研究所 松浦幹太准教授に心から感謝致します. 松浦准教授には, 研究の進め方や考え方, 研究に対する姿勢や着眼点について様々なご教授をして頂いただけでなく, 自身の論文の執筆に関する適切な助言や, 研究生活を送る上でのアドバイスまで, 多岐にわたってご指導して頂き, 修士 2 年間の研究生活は非常に充実したものとなりました. また学会参加や ISS square への参加, 企業との共同研究など数多くの活動の機会を与えて頂いたことで, あらゆる面で飛躍的に成長したと実感することができました. 改めて深く感謝致します.

また, 松浦研究室の定期ミーティングで, 研究内容に関して的確な助言を下さったり, 多くの鋭い質問をして頂いた, 中央大学の北川隆さん, 警察庁情報技術解析課の岡田智明さん, 情報処理推進機構の野島良さん, 産業技術研究所の田沼均さん, 中央大学の笠松宏平さんを始めとする, お世話になった全ての松浦研究室の定期ミーティングの参加者の方々に感謝致します.

そして, 松浦研究室との共同研究として, 主に技術的な指導をして下さった NTT プラットフォーム研究所の小林鉄太郎さん, 永井彰さんを始めとする共同研究の参加者の皆様にも, 改めて感謝致します.

さらに, 私たちの研究活動が円滑に進むように日頃から尽力してくださっている教授室秘書の小倉さん, 元教授室秘書の橋詰さんにも改めて深く感謝致します.

また, 松浦研究室の技術職員である細井琢朗さん, 松浦研究室メンバーである楊鵬さん, Jacob Schuldt さん, 松田隆宏さん, 中井泰雅さん, 施吃さん, Bongkot Jenjarrussakul さん, 市川顕君, 崔永錫さん, 付紹静さんにも, 日頃から研究室内での議論や, 松浦研究室の定期ミーティングにおいて, 活発に議論をしたり, 適切な助言をいただきました. 特に, 自身の投稿中の国際会議論文の共著となっている松田さん, Jacob さんには, 研究内容について基礎的な知識から高度な専門知識まで様々なご指導をして頂きました. 改めて深く感謝致します. 皆様のおかげで非常に充実した素晴らしい研究室生活を過ごすことができました.

最後に, 常日頃から私を支えてくれた家族に心から感謝します.

参考文献

[1] Digital Signature Standard (DSS). FIPS 186, 1994.

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

[3] M. Abe, Y. Cui, H. Imai, and E. Kiltz. Efficient hybrid encryption from ID-based encryption. *Designs, Codes and Cryptography*, 54(3):205–240, 2010.

[4] M. Abe, R. Gennaro, and K. Kurosawa. Tag-KEM/DEM: A new framework for hybrid encryption. *J. of Cryptology*, 21(1):97–130, 2008.

[5] S. Agrawal, D. Boneh, and X. Boyen. Lattice basis delegation in fixed dimension and shorter-ciphertext hierarchical IBE. In *Proc. of CRYPTO 2010*, volume 6223 of *LNCS*, pages 98–115. Springer, 2010.

[6] J. An, Y. Dodis, and T. Rabin. On the security of joint signature and encryption. In *Proc. of EUROCRYPT 2002*, volume 2332 of *LNCS*, pages 83–107. Springer, 2002.

[7] J. Baek, R. Steinfeld, and Y. Zheng. Formal proofs for the security of signcryption. In *Proc. of PKC 2002*, volume 2274 of *LNCS*, pages 80–98. Springer, 2002.

[8] J. Baek, R. Steinfeld, and Y. Zheng. Formal proofs for the security of signcryption. *J. of Cryptology*, 20(2):203–235, 2007.

[9] M. Bellare and C. Namprempre. Authenticated encryption: Relations among notions and analysis of the generic composition paradigm. In *Proc. of ASIACRYPT 2000*, volume 1976 of *LNCS*, pages 531–545. Springer, 2000.

[10] M. Bellare and P. Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In *Proc. of CCS 1993*, pages 62–73. ACM, 1993.

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

[12] D. Boneh and X. Boyen. Secure identity based encryption without random oracles. In *Proc. of CRYPTO 2004*, volume 3152 of *LNCS*, pages 443–459. Springer, 2004.

[13] D. Boneh, R. Canetti, S. Halevi, and J. Katz. Chosen-ciphertext security from identity-based encryption. *SIAM J. Computing*, 36(5):1301–1328, 2007.

[14] D. Boneh, E. Shen, and B. Waters. Strongly unforgeable signatures based on computational Diffie-Hellman. In *Proc. of PKC 2006*, volume 3958 of *LNCS*, pages 229–240. Springer, 2006.

[15] X. Boyen, Q. Mei, and B. Waters. Direct chosen ciphertext security from identity-based techniques. In *Proc. of CCS 2005*, pages 320–329. ACM, 2005.

[16] X. Boyen, Q. Mei, and B. Waters. Direct chosen ciphertext security from identity-based techniques, 2005. Updated version of [15]. Cryptology ePrint Archive: Report 2005/288. http://eprint.iacr.org/2005/288/.

[17] J. Camenisch and A. Lysyanskaya. Signature schemes and anonymous credentials from bilinear maps. In *Proc. of CRYPTO 2004*, volume 3152 of *LNCS*, pages 56–72. Springer, 2004.

[18] R. Canetti, O. Goldreich, and S. Halevi. The random oracle methodology, revisited. In *Proc. of STOC 1998*, pages 209–218. ACM, 1998.

[19] J. Cheon, N. Hopper, Y. Kim, and I. Osipkov. Provably secure timed-release public key encryption. volume 11, 2008.

[20] A. Dent. Hybrid signcryption schemes with outsider security (extended abstract). In *Proc. of ISC 2005*, volume 3650 of *LNCS*, pages 203–217. Springer, 2005.

[21] W. Diffie and M. Hellman. New directions in cryptography. In *Information Theory*, volume 22, pages 644–654, 1976.

[22] Y. Dodis and J. Katz. Chosen-ciphertext security of multiple encryption. In *Proc. of TCC 2005*, volume 3650 of *LNCS*, pages 188–209. Springer, 2005.

[23] A. Fiat and A. Shamir. How to prove yourself: Practical solutions to identification and signature problems. In *CRYPTO*, pages 186–194, 1986.

[24] M. C. Gorantla, C. Boyd, and J. M. G. Nieto. On the connection between signcryption and one-pass key establishment. In *IMA Int. Conf.*, pages 277–301, 2007.

[25] G. Hanaoka and K. Kurosawa. Efficient chosen ciphertext secure public key encryption under the computational diffie-hellman assumption. In *Proc. of ASIACRYPTO 2008*, volume 5350 of *LNCS*, pages 308–325. Springer, 2008.

[26] D. Hofheinz and E. Kiltz. Practical chosen ciphertext secure encryption from factoring. In *Proc. of EUROCRYPT 2009*, volume 5479 of *LNCS*, pages 313–332. Springer, 2009.

[27] S. Hohenberger and B. Waters. Short and stateless signatures from the RSA assumption. In *Proc. of CRYPTO 2009*, volume 5677 of *LNCS*, pages 654–670. Springer, 2009.

[28] I. R. Jeong, H. Y. Jeong, H. S. Rhee, D. H. Lee, and J. I. Lim. Provably secure encrypt-then-sign composition in hybrid signcryption. In *ICISC*, pages 16–34, 2002.

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

[30] E. Kiltz. Chosen-ciphertext secure key-encapsulation based on gap hashed diffie-hellman. In *Proc. of PKC 2007*, volume 4450 of *LNCS*, pages 282–297. Springer, 2007.

[31] C. Li, G. Yang, D. Wang, X. Deng, and S. Chow. An efficient signcryption scheme with key privacy. In *Proc. of EuroPKI 2007*, volume 4582 of *LNCS*, pages 78–93. Springer, 2007.

[32] B. Libert and J.-J. Quisquater. Improved signcryption with key privacy from gap Diffie-Hellman groups. In *Proc. of PKC 2004*, volume 2947 of *LNCS*, pages 187–200. Springer, 2004.

[33] B. Libert and J.-J. Quisquater. Improved signcryption with key privacy from gap Diffie-Hellman groups, 2004. Updated version of [32]. Avaliable at http://www.dice.usl.ac.be/ libert/.

[34] C. Ma. Efficient short signcryption schemewith public verifiability. In *Inscrypt*, pages 118–129, 2006.

[35] P. MacKenzie, M. Reiter, and K. Yang. Alternatives to non-malleability: Definitions, constructions and applications. In *Proc. of TCC 2004*, volume 2951 of *LNCS*, pages 171–190. Springer, 2004.

[36] T. Matsuda, K. Matsuura, and J. Schuldt. Efficient constructions of signcryption schemes and signcryption composability. In *Proc. of INDOCRYPT 2009*, volume 5922 of *LNCS*, pages 321–342. Springer, 2009.

[37] J. Nielsen. Separating random oracle proofs from complexity theoretic proofs: The non-committing encryption case. In *CRYPTO*, pages 111–126, 2002.

[38] D. Phan and D. Pointcheval. About the security of ciphers (semantic security and pseudo-random permutations). In *Proc. of SAC 2004*, volume 3357 of *LNCS*, pages 182–197. Springer, 2005.

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

[40] V. Shoup. A proposal for an iso standard for public key encryption (version 2.1). 2001. Avaliable at http://shoup.net/papers/.

[41] C. Tan. Security analysis of signcryption scheme from -diffie-hellman problems. *IEICE Transactions*, 89-A(1):206–208, 2006.

[42] C. Tan. Forgery of provable secure short signcryption scheme. *IEICE Transactions*, 90-A(9):1879–1880, 2007.

[43] C. Tan. Signcryption scheme in multi-user setting without random oracles. In *Proc. of IWSEC 2008*, volume 5312 of *LNCS*, pages 64–82. Springer, 2008.

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

[45] Y. Zheng. Digital signcryption or how to achieve cost(signature & encryption) << = cost(signature) + cost(encryption). In *Proc. of CRYPTO 1997*, volume 1294 of *LNCS*, pages 165–179. Springer, 1997.

発表文献

査読付き国際会議投稿済み審査中

i Daiki Chiba, Takahiro Matsuda, Schuldt C.N.Jacob, Kanta Matsuura. “Efficient Generic Constructions of Signcryption with Insider Security in the Multi-user Setting”, Proc. of 9th International Conference on Applied Cryptography and Network Security (ACNS ’11), Nerja (Malaga), Spain. June. 2011.

査読無し国内会議投稿論文

ii 千葉大輝, 松田隆宏, 松浦幹太. “タグベース KEM の選択的タグ安全性から適応的タグ安全性へのカメレオンハッシュを用いた強化手法と Signcryption への応用”, 2010 年 暗号と情報セキュリティシンポジウム (SCIS 2010) 予稿集 CDROM, 3A2-1. 高松, 1 月・2010 年.

iii 千葉大輝, 松田隆宏, シュルツ・ヤコブ, 松浦幹太. “多人数モデルで内部者安全な Signcryption の一般的構成法”, 2011 年 暗号と情報セキュリティシンポジウム (SCIS 2011) 予稿集 CDROM, 2A4-4. 小倉, 1 月・2011 年.

研究会発表

iv 千葉大輝. “タイムリリース暗号アプリケーションのインターフェース仕様について”, 第 12 回ペアリングフォーラム, 2010. [1]

v 千葉大輝. “タイトル未定”, 第 13 回ペアリングフォーラム, 2011(発表予定).

[1] “ペアリングフォーラム” での発表は, 本研究の成果を直接は含まないが, 当該分野で本来は一般的構成法の意義が大きいタイプの実装プロジェクトに取り組んだ関連発表である.