

選択暗号文攻撃に対し安全な公開鍵暗号の平文空間を拡げる一方法

松田 隆宏 花岡 悟一郎

産業技術総合研究所 セキュアシステム研究部門
305-8568 茨城県つくば市梅園 1-1-1 つくば中央第2事業所つくば本部・情報技術共同研究棟
{t-matsuda,hanaoka-goichiro}@aist.go.jp

あらまし 平文空間が1ビットの選択暗号文攻撃 (CCA) に対し安全な公開鍵暗号 (PKE) から任意長の平文を暗号化できる CCA 安全な PKE (または CCA 安全な鍵カプセル化メカニズム (KEM)) を構成する方法は、Myers と Shelat (FOCS'09) が可能であることを示して以降、Hohenberger ら (EUROCRYPT'12) 及び Matsuda と Hanaoka (IWSEC'13) によりさらなる簡素化・効率化が図られた。本稿では、後者の2つの研究において用いられた“検出可能 PKE” (及び KEM) を構成要素とした CCA 安全な KEM の新たな構成法を示す。本稿で示す構成法により 1 ビット PKE から CCA 安全な KEM を構成した場合、(違いはわずかながら) これまで知られるどの方法よりも暗号文サイズを小さくできる。

Yet Another Method for Expanding the Plaintext Space of Chosen Ciphertext Secure Public Key Encryption

Takahiro Matsuda Goichiro Hanaoka

Research Institute for Secure Systems (RISEC), AIST,
Tsukuba Central 2, 1-1-1, Umezono, Tsukuba, Ibaraki 305-8568.

Abstract The problem of how to construct a chosen ciphertext secure (CCA secure) public key encryption (PKE) scheme that can encrypt arbitrarily long plaintexts (or equivalently a CCA secure key encapsulation mechanism (KEM)), from CCA secure PKE with 1-bit plaintext space, was first resolved affirmatively by Myers and Shelat (FOCS'09), and the subsequent works by Hohenberger et al. (EUROCRYPT'12) and Matsuda and Hanaoka (IWSEC'13) showed simpler and more efficient constructions. In this paper, we show yet another method for constructing a CCA secure KEM from 1-bit PKE, based on the notion of “detectable PKE” (and KEM) that was studied and used in the latter two works. The new construction of a CCA secure KEM from 1-bit PKE that is obtained via our method has (slightly) smaller ciphertext size than the current best constructions.

1 はじめに

本稿では、平文空間が1ビット公開鍵暗号 (PKE) から、任意の長さの平文を暗号化できる PKE を構成する方法について再考する。選択平文攻撃 (CPA) に対する安全性や、非適応的選択暗号文攻撃 (CCA1) に対する安全性においては、最も簡素な“連結”構成、すなわち、暗号化したい平文を構成要素となる PKE が許す長さに分割し、各ブロック毎に暗号化してできた暗号文を連結する構成を用いれば、平文空間を拡げることができることはよく知られている。

しかし、この“連結”構成には適応的選択暗号文攻撃 (本稿では単に CCA と書く) が示せてしまう。

CCA 安全な PKE の平文空間を拡げることができるか、という問題は、比較的最近まで未解決問題であったが、Myers と Shelat [11] が¹、CCA 安全な1ビット PKE¹ から CCA 安全な鍵カプセル化メカニズム (KEM) を構成できることを示したことによって肯定的に解決された。² しか

¹以降、平文空間が1ビットの PKE を“1ビット PKE”と呼ぶ。

²CCA 安全な KEM が構成できれば、CCA 安全な共通鍵

し、最終的に得られる KEM の構成は非常に複雑であった。

CCA 安全な 1 ビット PKE から CCA 安全な KEM を構成する方法は、後に Hohenberger ら [6]、及び Matsuda と Hanaoka [8] によって簡素化・効率化が図られた。特に、[6] では、“検出可能 PKE” (Detectable PKE, DPKE) とその安全性定義として “検出可能 CCA 安全性” (DCCA 安全性) と “予測不可能性” (Unpredictability) が定式化され、これらの安全性を満たす 1 ビット DPKE を用いた CCA 安全な KEM の構成が示された。[8] はハイブリッド暗号の手法を用いて [6] の構成法を効率化した。

過去の構成法 [11, 6, 8] は、構成要素の違いはあるが、全て “二層暗号化” の手法を用いて構成されている。この手法では、平文は “内側暗号化” と “外側暗号化” により二重に (二層に) 暗号化される。より具体的には、平文 (あるいは KEM であればセッション鍵) と “外側暗号化” 用の乱数を “内側暗号化” 用の鍵で暗号化して “内側暗号文” を作成し、その内側暗号文をさらに “外側暗号化” 用の鍵で、内側暗号文に含まれる乱数を用いて暗号化する、という構成になっている。そして、1 ビット PKE から内側暗号化用と外側暗号化用の暗号方式をそれぞれ構成することで、全体として 1 ビット PKE から CCA 安全な PKE/KEM が構成される。

本稿では、“二層暗号化” の手法による 1 ビット PKE から CCA 安全な KEM の構成について、さらなる簡素化・効率化が可能かどうかについて取り組む。

本研究の貢献 本稿では主成果として、一般的な暗号要素技術を、[11, 6, 8] の様に “二層暗号化” 構成の手法で組み合わせる新たな CCA 安全な KEM の構成法を示す。具体的には、3 節において、“内側暗号化” には [6, 8] と同じく検出可能 PKE (より正確には KEM) を、“外側暗号化” にはコミットメント方式と穴あけ可能タグベース暗号 (Puncturable Tag-Based Encryption, PTBE) [10] を組み合わせることで、各構成要素が適切な安全性を満たすという仮定の下、“二層暗号化” 構成の CCA 安全性を証明できることを示す。さらに、4 節において、提案方式の各要素技術を、CCA 安全な 1 ビット PKE を用いて既存の最も効率の良い方法で実現した場合、既存の最も効率の良い構成法と比べ、(違いはわずかながら) 暗号文サイズを小さくできる

暗号 (SKE) と組み合わせることで、任意長の平文を暗号化可能な CCA 安全な PKE を構成できる [2]。また、CCA 安全な SKE は、一方向関数などから構成できるため、当然 CCA 安全な PKE から構成できる [15]。

ことも説明する。

今後の課題 PTBE は、CPA 安全な PKE と非対話型ゼロ知識証明 (NIZK) を用いて CCA 安全な PKE を構成する “Dolev-Dwork-Naor” (DDN) 構成の核となる部分を抽象化した暗号要素技術である。従って提案構成は、“二層暗号化” 構成の外側暗号化として、DDN 構成の手法を用いることができることを示している。これに対し、[6, 8] では、1-CCA 安全³ な PKE と CPA 安全な PKE とを、“Naor-Yung” 構成 [14] の様な “二重暗号化” の手法を用いて組み合わせている。このため、提案手法と [6, 8] の構成とは、それらの “外側暗号化” の構成要素が、“PTBE とコミットメント方式を組み合わせた DDN 構成の手法” と “1-CCA 安全な PKE と CPA 安全な PKE を組み合わせた NY 構成の手法” のトレードオフを示すような関係になっている。この関係は理論的には興味深い事実であるが、あくまでも定性的な説明であり、提案構成と [6, 8] の構成法の間の何らかの何らかの定量的な関係を明らかにすることを今後の課題としたい。

また、本稿では、CCA 安全な 1 ビット PKE から CCA 安全な KEM を構成するこれまでよりも効率的な構成法を示すが、漸近的な意味では効率は既存方式と差が無い。より簡潔で効率的な構成法が可能か (または不可能か) について明らかにすることも今後の課題としたい。

2 準備

本節では、表記法や要素技術の定義を行う。

基礎的な表記法 $\mathbb{N}$ は自然数の集合を指す。 $n \in \mathbb{N}$ ならば、 $[n] = \{1, \dots, n\}$ とする。 x と y がビット列ならば、“ $|x|$ ” は x のビット長を表し、“ $x||y$ ” は x と y を連結したビット列を表す。 S が集合ならば、“ $x \leftarrow S$ ” は S から要素を一様ランダムに取り出し x に代入する操作を表す。“PPTA” は確率的多項式時間アルゴリズムを指す。 $\mathcal{A}$ が確率的アルゴリズムならば、“ $y \leftarrow \mathcal{A}(x; r)$ ” は、 $\mathcal{A}$ が x を入力とし、 r を乱数として実行され、 y を出力する操作を表し、 $\mathcal{A}$ が用いた乱数を明示する必要が無い場合は単に “ $y \leftarrow \mathcal{A}(x)$ ” と書く。“ $\mathcal{A}^{\mathcal{O}}$ ” は $\mathcal{A}$ がオラクル $\mathcal{O}$ にアクセスできることを表す。“ k ” は常にセキュリティパラメータを指すことにする。関数 $f: \mathbb{N} \rightarrow [0, 1]$ が全ての非負の多項式 $p(k)$ と十分大きな全ての k について $f(k) < 1/p(k)$ ならば、 f を無視できるという (negligible 関数)。

³一回の復号クエリを行う攻撃者に対する安全性 [1]。

2.1 鍵カプセル化メカニズム (KEM)

$n = n(k) > 0$ を多項式とする。セッション鍵空間が $\{0, 1\}^n$ である様な KEM Γ は、以下の様なインターフェースの 3 つの PPTA (KKG, Encap, Decap) からなる:

$$\begin{aligned} \text{鍵生成: } (pk, sk) &\leftarrow \text{KKG}(1^k) \\ \text{カプセル化: } (c, K) &\leftarrow \text{Encap}(pk) \\ \text{復号: } K / \perp &\leftarrow \text{Decap}(sk, c) \end{aligned}$$

ただし上記において、Decap は確定的アルゴリズム、 (pk, sk) は公開鍵/秘密鍵対、 c はセッション鍵 $K \in \{0, 1\}^n$ の pk の下での暗号文である。

以下の“正当性”が要求される: “全ての $k \in \mathbb{N}$ 、 $(pk, sk) \leftarrow \text{KKG}(1^k)$ 、及び $(c, K) \leftarrow \text{Encap}(pk)$ について、 $\text{Decap}(sk, c) = K$ ”

検出可能 KEM (DKEM) PPTA の 4 つ組 $\Gamma = (\text{KKG}, \text{Encap}, \text{Decap}, F)$ が以下を満たすとき、 Γ を“検出可能 KEM”(Detectable KEM, DKEM) [6, 8] であるという:

- (1) $(\text{KKG}, \text{Encap}, \text{Decap})$ は KEM である。
- (2) F は確定的アルゴリズムであり、公開鍵 pk と 2 つの暗号文 (c^*, c') を入力として受け取り、0 または 1 を出力する。

F は、 Γ の“KEM としての機能”には何も影響を与えないが、安全性の定義において重要な役割を果たす。直観的には、 F は「安全性ゲームにおいてチャレンジ暗号文が c^* のときに、復号クエリ c' に答えることは“危険”である」ことを“検出”するための述語であり、それが要素技術の名前の由来である。

(D)KEM の安全性 ここでは KEM の CCA 安全性、及び DKEM の DCCA 安全性と予測不可能性 (Unpredictability) を振り返る。

- $\text{ATK} \in \{\text{CCA}, \text{DCCA}\}$ とする。全ての PPTA 攻撃者 $\mathcal{A}$ に対し、以下で定義される $\text{Adv}_{\Gamma, \mathcal{A}}^{\text{ATK}}(k)$ が無視できる場合、(D)KEM Γ は ATK 安全であるという:

$$\begin{aligned} \text{Adv}_{\Gamma, \mathcal{A}}^{\text{ATK}}(k) &:= 2 \cdot |\Pr[(pk, sk) \leftarrow \text{KKG}(1^k); \\ &(c^*, K_1^*) \leftarrow \text{Encap}(pk); K_0^* \leftarrow \{0, 1\}^k; b \leftarrow \{0, 1\}; \\ &b' \leftarrow \mathcal{A}^{\text{Decap}(sk, \cdot)}(pk, c^*, K_b^*) : b' = b] - \frac{1}{2}| \end{aligned}$$

ただし、 $\text{ATK} = \text{CCA}$ の場合、 $\mathcal{A}$ は c^* をクエリしてはならない。また、 $\text{ATK} = \text{DCCA}$ の場合、 $\mathcal{A}$ は $F(pk, c^*, c) = 1$ を満たす c をクエリしてはならない。

- 全ての PPTA 攻撃者 $\mathcal{A}$ に対し、以下で定義される $\text{Adv}_{\Gamma, \mathcal{A}}^{\text{UNP}}(k)$ が無視できる場合、DKEM Γ は予測不可能性を満たすという:

$$\begin{aligned} \text{Adv}_{\Gamma, \mathcal{A}}^{\text{UNP}}(k) &:= \Pr[(pk, sk) \leftarrow \text{KKG}(1^k); \\ c' &\leftarrow \mathcal{A}^{\text{Decap}(sk, \cdot)}(pk); (c^*, K^*) \leftarrow \text{Encap}(pk) : \\ &F(pk, c^*, c') = 1] \end{aligned}$$

2.2 穴あけ可能タグベース暗号 (PTBE)

タグベース暗号 (TBE) [7] は PKE の一般化であり、暗号化・復号時には各鍵に加えタグ tag が使用される。暗号文は、暗号化時に用いたものと同一のタグを用いて復号できる。

“穴あけ可能”タグベース暗号 (Puncturable TBE, PTBE) [10] は、直観的に言えば、タグ空間に一点の“穴”を開けることができる TBE である。より正確に言えば、秘密鍵 sk と“穴”としたいタグ tag^* から“穴あき秘密鍵” $\widehat{sk}_{\text{tag}^*}$ を生成できる。 $\widehat{sk}_{\text{tag}^*}$ は、“穴”以外のタグを用いて生成された暗号文を復号できる反面、“穴” tag^* を用いて作成された暗号文から情報を得るためには役に立たない。

シンタックス PTBE $\mathcal{T}$ は以下の様な入出力を持つ 5 つのアルゴリズム (TKG, TEnc, TDec, Punc, $\widehat{\text{TDec}}$) からなる⁴:

$$\begin{aligned} \text{鍵生成: } (pk, sk) &\leftarrow \text{TKG}(1^k) \\ \text{暗号化: } c &\leftarrow \text{TEnc}(pk, \text{tag}, m) \\ \text{復号: } m / \perp &\leftarrow \text{TDec}(sk, \text{tag}, c) \\ \text{秘密鍵穴あけ: } \widehat{sk}_{\text{tag}^*} &\leftarrow \text{Punc}(sk, \text{tag}^*) \\ \text{穴あき復号: } m / \perp &\leftarrow \widehat{\text{TDec}}(\widehat{sk}_{\text{tag}^*}, \text{tag}, c) \end{aligned}$$

ただし上記において、 (pk, sk) は公開鍵/秘密鍵対、 c は平文 m の pk とタグ $\text{tag} \in \{0, 1\}^k$ の下での暗号文である。 $\widehat{sk}_{\text{tag}^*}$ は“穴” $\text{tag}^* \in \{0, 1\}^k$ を持つ穴あき秘密鍵であり、 $\widehat{\text{TDec}}$ は $\widehat{sk}_{\text{tag}^*}$ を用いて復号するためのアルゴリズムである。

PTBE の正当性として、以下が成り立つことが要求される: 全ての $k \in \mathbb{N}$ 、全ての $(pk, sk) \leftarrow \text{TKG}(1^k)$ 、全ての $\text{tag} \neq \text{tag}^*$ を満たす $\text{tag}, \text{tag}^* \in \{0, 1\}^k$ 、全ての平文 m 、及び全ての $c \leftarrow \text{TEnc}(pk, \text{tag}, m)$ について

$$\begin{aligned} \text{TDec}(sk, \text{tag}, c) &= \widehat{\text{TDec}}(\text{Punc}(sk, \text{tag}^*), \text{tag}, c) \\ &= m \end{aligned}$$

上記は、TEnc によって生成された暗号文 c についてしか保証されない点に注意されたい。

⁴本稿では、PTBE のタグ空間を $\{0, 1\}^k$ とする。

PTBEの安全性 ここでは、[10]で定式化された“拡張”(Extended) CPA 安全性 (eCPA 安全性) の定義を振り返る。

全ての PPTA 攻撃者 $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ に対し、以下で定義される $\text{Adv}_{\mathcal{T}, \mathcal{A}}^{\text{eCPA}}(k)$ が無視できる場合、PTBE $\mathcal{T}$ は eCPA 安全であるという:

$$\begin{aligned} \text{Adv}_{\mathcal{T}, \mathcal{A}}^{\text{eCPA}}(k) &:= 2 \cdot \Pr[(\text{tag}^*, m_0, m_1, \text{st}) \leftarrow \mathcal{A}_1(1^k); \\ & (pk, sk) \leftarrow \text{TKG}(1^k); \widehat{sk}_{\text{tag}^*} \leftarrow \text{Punc}(sk, \text{tag}^*); \\ & b \leftarrow \{0, 1\}; c^* \leftarrow \text{TEnc}(pk, \text{tag}^*, m_b); \\ & b' \leftarrow \mathcal{A}_2(\text{st}, pk, c^*, \widehat{sk}_{\text{tag}^*}) : b' = b] - \frac{1}{2} \end{aligned}$$

ただし $|m_0| = |m_1|$ でなければならない。

具体的な構成法 PTBE は元来、DDN 構成 [4] の構成の核となる部分を抽象化⁵する目的で導入されたものであり、任意の CPA 安全な PKE から構成できる [10]。[10] の構成では、タグ空間と平文空間が $\{0, 1\}^k$ の PTBE を、平文空間が $\{0, 1\}^k$ の CPA 安全な PKE から構成する場合、PTBE としての鍵生成時には $2k$ 個の鍵対を生成する。また、暗号化時には、 $2k$ 個の公開鍵からタグの値に従って k 個選び出し、選ばれた各鍵でそれぞれ独立に平文を暗号化して得られた k 個の暗号文を連結したものを、PTBE としての暗号文とする。

k ビットよりも長い平文を暗号化するためには、PTBE で k ビットの乱数 K を暗号化し、その乱数 K を SKE の鍵として実際の平文を暗号化する、という典型的なハイブリッド暗号の手法を用いることができる。その際、SKE には、一回の暗号化クエリの下での識別不可能性という、非常に弱い安全性のみ満たすものを用いればよい。この様な SKE は、例えば (鍵 K をシードとした) 擬似乱数生成器 (PRG) の出力と平文の排他的論理和を行う構成で実現できる。

2.3 コミットメント方式

3 節で示す提案方式では、信頼できるセットアップを許す非対話型のコミットメント方式を用いるため、ここではその定義を振り返る。

コミットメント方式 $\mathcal{C}$ は、以下の様なインタフェースの PPTA の組 (CKG, Com) からなる:

$$\begin{aligned} &\text{鍵生成: } ck \leftarrow \text{CKG}(1^k) \\ &\text{コミットメント生成: } c \leftarrow \text{Com}(ck, m) \end{aligned}$$

ただし上記において、 ck はコミットメント鍵、 c は平文 m の ck の下でのコミットメントである。

⁵より具体的には、DDN 構成から、非対話型ゼロ知識証明と電子署名を除いた部分を、TBE として定式化した要素技術である。

コミットメント方式の安全性 ここでは、本稿で用いるコミットメント方式に要求する安全性定義を振り返る。⁶

- 全ての PPTA 攻撃者 $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ に対し、以下で定義される $\text{Adv}_{\mathcal{C}, \mathcal{A}}^{\text{Hide}}(k)$ が無視できる場合、コミットメント方式 $\mathcal{C}$ は秘匿性を満たすという:

$$\begin{aligned} \text{Adv}_{\mathcal{C}, \mathcal{A}}^{\text{Hide}}(k) &:= 2 \cdot \Pr[(m_0, m_1) \leftarrow \mathcal{A}_1(1^k); \\ & ck \leftarrow \text{CKG}(1^k); b \leftarrow \{0, 1\}; c^* \leftarrow \text{Com}(ck, m_b) \\ & b' \leftarrow \mathcal{A}_2(\text{st}, ck, c^*) : b' = b] - \frac{1}{2} \end{aligned}$$

ただし、 $|m_0| = |m_1|$ でなければならない。

- 全ての PPTA 攻撃者 $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ に対し、以下で定義される $\text{Adv}_{\mathcal{C}, \mathcal{A}}^{\text{Bind}}(k)$ が無視できる場合、コミットメント方式 $\mathcal{C}$ はターゲット束縛性を満たすという:

$$\begin{aligned} \text{Adv}_{\mathcal{C}, \mathcal{A}}^{\text{Bind}}(k) &:= \Pr[(m, r, \text{st}) \leftarrow \mathcal{A}_1(1^k); \\ & ck \leftarrow \text{CKG}(1^k); (m', r') \leftarrow \mathcal{A}_2(\text{st}, ck) : \\ & \text{Com}(ck, m'; r') = \text{Com}(ck, m; r) \wedge m' \neq m] \end{aligned}$$

上記の“秘匿性”と“束縛性”の定義は、以下の意味で通常同要素技術に要求される秘匿性・束縛性よりも弱い: 秘匿性の安全性試行では、攻撃者はコミットメント鍵 ck を与えられる前にチャレンジ平文を決定せねばならない。また、束縛性の安全性試行では、攻撃者は片方の平文/乱数対をコミットメント鍵 ck を与えられる前に決定せねばならない。後者は、(鍵付き) ハッシュ関数のターゲット衝突困難性 (汎用一方向性とも呼ばれる [13]) の様な安全性であるため、本稿ではターゲット束縛性と呼ぶ。

コミットメントのサイズに関する要件と具体的な構成法の例 3 節で示す提案構成では、コミットメントの値を PTBE のタグとして用いる必要がある。このため、提案方式のために必要なコミットメント方式では、機能的な要件として、コミットする平文のサイズに依らず、コミットメントのサイズが k ビットであることを要求する。この様なコミットメント方式は、例えば CPA に対し安全な PKE で平文を暗号化し、暗号文をターゲット衝突困難ハッシュ関数に通して得られるハッシュ値をコミットメントとする、という構成で達成できる。(コミットメント鍵は、CPA 安全な PKE の公開鍵と、ターゲット衝突困難ハッシュ関数のハッシュ鍵からなる。)

⁶ここで使用する安全性定義は、[10]で用いられた定義と同様である。

また、本稿ではコミットメント鍵に依存した平文に対する秘匿性を考えないため、コミットメントフェーズが2メッセージ型のコミットメント方式において、受信者から送信者への最初のメッセージをコミットメント鍵とみなす方式も用いることができる。従って例えば、Naor [12] のコミットメント方式を用いることができる。(ただし、コミットメントのサイズに関する要求を達成するためには、上記の方式同様、ターゲット衝突困難ハッシュを用いる必要がある。)

3 提案構成

本節では、本稿の主成果である、“二層暗号化”による新たな CCA 安全な KEM の構成を示す。

$\Gamma_{\text{in}} = (\text{KKG}_{\text{in}}, \text{Encap}_{\text{in}}, \text{Decap}_{\text{in}}, \text{F}_{\text{in}})$ を、暗号文サイズが $n = n(k)$ 、セッション鍵空間が $\{0, 1\}^{3k}$ である様な DKEM、 $\mathcal{C} = (\text{CKG}, \text{Com})$ を、平文空間が $\{0, 1\}^n$ 、コミットメントサイズが k ビット、 Com の乱数空間が $\{0, 1\}^k$ である様なコミットメント方式、 $\mathcal{T} = (\text{TKG}, \text{TEnc}, \text{TDec}, \text{Punc}, \widehat{\text{TDec}})$ をタグ空間が $\{0, 1\}^k$ 、平文空間が $\{0, 1\}^n$ 、 TEnc の乱数空間が $\{0, 1\}^k$ である様な PTBE であるとする。⁷

Γ_{in} 、 $\mathcal{C}$ 、及び $\mathcal{T}$ を構成要素として、KEM $\Gamma = (\text{KKG}, \text{Encap}, \text{Decap})$ を以下の様に構成する：

KKG(1^k): $(pk_{\text{in}}, sk_{\text{in}}) \leftarrow \text{KKG}_{\text{in}}(1^k)$ 、 $ck \leftarrow \text{CKG}(1^k)$ 、及び $(pk, sk) \leftarrow \text{TKG}(1^k)$ を実行する。公開鍵 $PK = (pk_{\text{in}}, ck, pk)$ と秘密鍵 $SK = (sk_{\text{in}}, sk, PK)$ を出力する。

Encap(PK): $(c_{\text{in}}, \alpha) \leftarrow \text{Encap}_{\text{in}}(pk_{\text{in}})$ を実行し、 $\alpha = (r_1, r_2, K) \in (\{0, 1\}^k)^3$ と分割する。 $\text{tag} \leftarrow \text{Com}(ck, c_{\text{in}}; r_1)$ 及び $c \leftarrow \text{TEnc}(pk, \text{tag}, c_{\text{in}}; r_2)$ を実行する。暗号文 $C = (\text{tag}, c)$ とセッション鍵 K を出力する。

Decap(SK, C): $C = (\text{tag}, c)$ と分割する。 $c_{\text{in}} \leftarrow \text{TDec}(sk, \text{tag}, c)$ を実行し、 $c_{\text{in}} = \perp$ ならば $\perp$ を出力して停止する。 $\alpha \leftarrow \text{Decap}_{\text{in}}(sk_{\text{in}}, c_{\text{in}})$ を実行し、 $\alpha = \perp$ ならば $\perp$ を出力して停止する。 $\alpha = (r_1, r_2, K) \in (\{0, 1\}^k)^3$ と分割する。 $\text{Com}(ck, c_{\text{in}}; r_1) = \text{tag}$ 及び $\text{TEnc}(pk, \text{tag}, c_{\text{in}}; r_2) = c$ が成り立つならばセッション鍵 K を、そうでなければ $\perp$ を出力する。

⁷DCCA 安全性かつ予測不可能性を満たす DKEM のセッション鍵空間は、PRG を用いることで、安全性を損なわずに自由に調節できる。また、 Com と TEnc の乱数空間が $\{0, 1\}^k$ であるという仮定も、一般性を失うものではない。これらのアルゴリズムに k ビットより長い乱数が必要だったとしても、 k ビットの PRG を用いて必要な長さに伸長してそれを用いればよい。(この様な変換によってコミットメント方式や PTBE の安全性は損なわれない。)

代替復号アルゴリズム [8, 9, 10] の様に、上記の構成の安全性証明では、以下で説明する“代替秘密鍵” $\widehat{SK}_{\text{tag}^*}$ と“代替復号アルゴリズム” AltDecap を考えるのが便利である：

$(PK, SK = (sk_{\text{in}}, sk, PK))$ を KKG から出力された鍵対、 $C^* = (\text{tag}^*, c^*)$ を $\text{Encap}(PK)$ から出力された暗号文とした時、 SK と C^* に対応した“代替秘密鍵” $\widehat{SK}_{\text{tag}^*}$ を $\widehat{SK}_{\text{tag}^*} = (\text{tag}^*, \widehat{sk}_{\text{tag}^*} = \text{Punc}(sk, \text{tag}^*), sk_{\text{in}}, PK)$ と定義する。また、“代替復号アルゴリズム” AltDecap を、代替秘密鍵 $\widehat{SK}_{\text{tag}^*}$ と暗号文 $C = (\text{tag}, c)$ を入力とし、以下の様に動作するアルゴリズムとする：

AltDecap($\widehat{SK}_{\text{tag}^*}, C$): $\text{tag} = \text{tag}^*$ の場合は $\perp$ を出力して停止する。そうでなければ、 c_{in} を、“ $c_{\text{in}} \leftarrow \text{TDec}(sk, \text{tag}, c)$ ”ではなく“ $c_{\text{in}} \leftarrow \widehat{\text{TDec}}(\widehat{sk}_{\text{tag}^*}, \text{tag}, c)$ ”として計算する以外は、 $\text{Decap}(SK, C)$ と同様に動作し、最後に復号結果 K (または $\perp$) を出力する。

上記の代替復号アルゴリズム AltDecap について、以下が成り立つ。

補題 1. (PK, SK) を $\text{KKG}(1^k)$ から出力された鍵対、 $C^* = (\text{tag}^*, c^*)$ を $\text{Encap}(PK)$ から出力された暗号文とし、 $\widehat{SK}_{\text{tag}^*}$ を SK と C^* に対応した代替秘密鍵とする。この時、 $(\text{Encap}$ の出力かどうかに限らず) 全ての暗号文 $C = (\text{tag}, c)$ について、 $\text{tag} \neq \text{tag}^*$ ならば $\text{Decap}(SK, C) = \text{AltDecap}(\widehat{SK}_{\text{tag}^*}, C)$ が成り立つ。

証明 $C = (\text{tag}, c)$ を $\text{tag} \neq \text{tag}^*$ を満たす任意の暗号文とする。また、表記の簡単化のため、 $c_{\text{in}} = \text{TDec}(sk, \text{tag}, c)$ 、 $c'_{\text{in}} = \widehat{\text{TDec}}(\widehat{sk}_{\text{tag}^*}, \text{tag}, c)$ とする。

補題を示すためには、 c_{in} 及び c'_{in} の値によって場合分けして考えればよい。 $c_{\text{in}} = c'_{\text{in}}$ のときは、 $\text{Decap}(SK, C)$ と $\text{AltDecap}(\widehat{SK}_{\text{tag}^*}, C)$ は、それぞれ内部で c_{in} と c'_{in} を計算した後は完全に同一の動作を行うので両者の出力は一致する。また、 $c_{\text{in}} \neq c'_{\text{in}}$ のときは、 $\text{Decap}(SK, C) = \text{AltDecap}(\widehat{SK}_{\text{tag}^*}, C) = \perp$ が成り立つ。PTBE の正当性により、 c が $\text{TEnc}(pk, \text{tag}, \cdot)$ の出力結果かつ $\text{tag} \neq \text{tag}^*$ であれば、 $c_{\text{in}} = c'_{\text{in}}$ が成り立つはずなので、“ $c_{\text{in}} \neq c'_{\text{in}}$ ”は、 c が $\text{TEnc}(pk, \text{tag}, \cdot)$ の正しい出力ではないことを意味する。ところが、 Decap 及び AltDecap の最後に実行される再暗号化による c のチェックは、 c が $\text{TEnc}(pk, \text{tag}, \cdot)$ の出力結果でなければ決して満たされることはない。従って Decap と AltDecap はいずれも C の復号結果として $\perp$ を出力する。なお、 $c_{\text{in}} = \perp$ または $\text{Decap}_{\text{in}}(sk_{\text{in}}, c_{\text{in}}) = \perp$ の場合も考えら

れるが、その場合はいずれにしても Decap は $\perp$ を出力する。 c'_{in} についても同様である。 $\square$

Γ の CCA 安全性 提案方式 Γ の CCA 安全性は、以下の様に示すことができる。

定理 1. $DKEM \Gamma_{\text{in}}$ が DCCA 安全性かつ予測不可能性を満たし、コミットメント方式 $\mathcal{C}$ が秘匿性とターゲット束縛性を満たし、 $PTBE \mathcal{T}$ が eCPA 安全性を満たすとする。この時、 $KEM \Gamma$ は CCA 安全である。

証明のスケッチ $\mathcal{A}$ を Γ の CCA 安全性に対する攻撃者とし、以下のゲーム列を考える (下記では、アスタリスク付きの文字は $\mathcal{A}$ のチャレンジ暗号文 $C^* = (\text{tag}^*, c^*)$ 作成に使用される値を表し、明示されない操作は、直前のゲームと同一とする。):

ゲーム 1: Γ の CCA 試行そのもの。

ゲーム 2: $\mathcal{A}$ の復号クエリ $C = (\text{tag}, c)$ が $\text{tag} = \text{tag}^*$ を満たす場合は、 $\perp$ で返答する。

ゲーム 3: $\mathcal{A}$ の復号クエリ $C = (\text{tag}, c)$ が $\text{tag} \neq \text{tag}^*$ を満たす場合は、 $\text{AltDecap}(\widehat{SK}_{\text{tag}^*}, C)$ で返答する。

ゲーム 4: $r_1^*, r_2^*, K_1^* \in \{0, 1\}^k$ として、 c_{in}^* に対応したセッション鍵 α^* から得られる値でなく、一様乱数を用いる。

ゲーム 5: tag^* を $\text{tag}^* \leftarrow \text{Com}(ck, 0^n)$ と作成する。

ゲーム 6: c^* を $c^* \leftarrow \text{TEnc}(pk, \text{tag}^*, 0^n)$ と作成する。

上記ゲームにおいて、 $\mathcal{A}$ の復号クエリ $C = (\text{tag}, c)$ が、 $\text{tag} \neq \text{tag}^*$ 、 $\widehat{\text{TDec}}(sk_{\text{tag}^*}, \text{tag}, c) = c_{\text{in}} \neq \perp$ 、かつ $F_{\text{in}}(pk_{\text{in}}, c_{\text{in}}^*, c_{\text{in}}) = 1$ を満たす場合、 C を“危険クエリ”と呼ぶ。ただし、 $sk_{\text{tag}^*} = \text{Punc}(sk, \text{tag}^*)$ とする。

各ゲーム $i \in [6]$ に対し、 S_i を $\mathcal{A}$ がチャレンジビット b の推測に成功するイベント、 D_i を $\mathcal{A}$ が少なくとも一つ危険クエリを復号オラクルにクエリするイベントとする。

$\mathcal{A}$ の CCA アドバンテージの上限は以下の様に見積もることができる:

$$\begin{aligned} \text{Adv}_{\Gamma, \mathcal{A}}^{\text{CCA}}(k) &= 2 \cdot |\Pr[S_1] - \frac{1}{2}| \\ &\leq 2 \cdot \sum_{i \in [2]} |\Pr[S_i] - \Pr[S_{i+1}]| + 2 \cdot |\Pr[S_3] - \frac{1}{2}| \\ &\leq 2 \cdot \sum_{i \in [2]} |\Pr[S_i] - \Pr[S_{i+1}]| \\ &\quad + 2 \cdot |\Pr[S_3 \wedge \overline{D_3}] - \frac{1}{2}| + \Pr[D_3] \end{aligned}$$

(不等式は右上に続く $\nearrow$)

(不等式は左下からの続き)

$$\begin{aligned} &\leq 2 \cdot \sum_{i \in [2]} |\Pr[S_i] - \Pr[S_{i+1}]| \\ &\quad + 2 \cdot |\Pr[S_3 \wedge \overline{D_3}] - \Pr[S_4 \wedge \overline{D_4}]| \\ &\quad + 2 \cdot |\Pr[D_3] - \Pr[D_4]| \\ &\quad + 2 \cdot |\Pr[S_4 \wedge \overline{D_4}] + \frac{1}{2} \Pr[D_4] - \frac{1}{2}| \\ &\quad + \sum_{i \in \{4, 5\}} |\Pr[D_i] - \Pr[D_{i+1}]| + \Pr[D_6] \end{aligned} \quad (1)$$

ただし二番目の不等式では $|\Pr[S_3 \wedge D_3] - \frac{1}{2} \Pr[D_3]| \leq \frac{1}{2} \Pr[D_3]$ を用いた。

以下に、不等式 (1) の各項が無視できることを順に説明する。

第一に、 $\mathcal{C}$ のターゲット束縛性により、 $|\Pr[S_1] - \Pr[S_2]|$ が無視できることを示せる。ゲーム 1 とゲーム 2 は、 $\mathcal{A}$ が $\text{tag} = \text{tag}^*$ かつ $\text{Decap}(SK, C) \neq \perp$ となる暗号文 C を復号クエリしない限りは同一であるため、 $|\Pr[S_1] - \Pr[S_2]|$ は $\mathcal{A}$ がゲーム 1 (またはゲーム 2) で上記のようなクエリを行う確率で上限できる。ところが、 $\text{tag} = \text{tag}^*$ かつ $\text{Decap}(SK, C) \neq \perp$ を満たす復号クエリ $C = (\text{tag}, c)$ では、 $\text{TDec}(sk, \text{tag}, c) = c_{\text{in}} \neq c_{\text{in}}^*$ が成り立つ。 $(\text{tag} = \text{tag}^* \text{ と } \text{Decap}(SK, C) \neq \perp \text{ に加え } c_{\text{in}} = c_{\text{in}}^* \text{ が成り立つ場合は } C = C^* \text{ が成り立ち、} \mathcal{A} \text{ の復号クエリの条件 } C \neq C^* \text{ に矛盾するため。})$ よって、このような復号クエリはコミットメント方式のターゲット束縛性によって発見することは困難である。

第二に、 $|\Pr[S_2] - \Pr[S_3]| = 0$ である。ゲーム 2 とゲーム 3 の違いは、 $\mathcal{A}$ の復号クエリ $C = (\text{tag}, c)$ のうち $\text{tag} \neq \text{tag}^*$ を満たすものに対する応答であるが、補題 1 により、このような復号クエリに対する応答は完全に同一である。従って、 $\mathcal{A}$ にとってゲーム 2 とゲーム 3 は完全に同一のゲームであり、成功確率は両ゲームで等しい。

第三に、 Γ_{in} の DCCA 安全性により、 $|\Pr[S_3 \wedge \overline{D_3}] - \Pr[S_4 \wedge \overline{D_4}]|$ と $|\Pr[D_3] - \Pr[D_4]|$ が無視できることを示せる。次のような帰着アルゴリズム $\mathcal{B}$ を考える: $\mathcal{B}$ は pk_{in} とチャレンジ暗号文/セッション鍵対 $(c_{\text{in}}^*, \alpha_\sigma^*)$ を最初に受け取り (σ は $\mathcal{B}$ のチャレンジビット)、 $ck \leftarrow \text{CKG}(1^k)$ 及び $(pk, sk) \leftarrow \text{TKG}(1^k)$ を実行し、 $\alpha_\sigma^* = (r_1^* \| r_2^* \| K_1^*)$ を用いてゲーム 3 の様に $\mathcal{A}$ のチャレンジ暗号文/セッション鍵対 $(C^* = (\text{tag}^*, c^*), K_b^*)$ を作成し、 $\mathcal{A}(PK = (pk_{\text{in}}, ck, pk), C^*, K_b^*)$ を実行する。 $\mathcal{A}$ からの復号クエリ C に対し、もし C が危険クエリの場合は、 $\mathcal{B}$ は 0 を出力して停止する。(危険クエリかどうかの判定は、 $\widehat{sk}_{\text{tag}^*} = \text{Punc}(sk, \text{tag}^*)$ を用いて行うことができる。) 危険クエリでなければ、 $\mathcal{B}$ は $\text{AltDecap}(\widehat{SK}_{\text{tag}^*}, C)$ を $\widehat{sk}_{\text{tag}^*}$ と復

号オラクルを用いて計算できるため、 $\mathcal{A}$ のクエリに対し完全に応答できる。 $\mathcal{A}$ が最後まで危険クエリを行わずに推測ビットを出力して停止した際は、 $\mathcal{B}$ は $\mathcal{A}$ の推測が正しければ1を、そうでなければ0を出力する。

上記の $\mathcal{B}$ は $\mathcal{A}$ に対し、“ $\mathcal{A}$ が危険クエリを行うまで”はゲーム $(4-\sigma)$ を完全にシミュレートする。よって、 $\mathcal{B}$ が1を出力する確率、すなわち“ $\mathcal{A}$ が危険クエリを行わずにチャレンジビットの推測に成功する確率”は $\Pr[S_{4-\sigma} \wedge \overline{D_{4-\sigma}}]$ となる。つまり、 $\mathcal{B}$ のDCCAアドバンテージは、 $|\Pr[S_3 \wedge \overline{D_3}] - \Pr[S_4 \wedge \overline{D_4}]|$ となり、 Γ_{in} のDCCA安全性により無視できる。また、 $\mathcal{B}$ と同様に動作するが、 $\mathcal{A}$ が危険クエリを行った場合のみ1を出力する攻撃者 $\mathcal{B}'$ を考えると、そのDCCAアドバンテージは $|\Pr[D_3] - \Pr[D_4]|$ であり、同様に無視できる。

第四に、 $|\Pr[S_4 \wedge \overline{D_4}] + \frac{1}{2}\Pr[D_4] - \frac{1}{2}| = 0$ である。ゲーム4ではチャレンジ暗号文 C^* が K_1^* とは独立となり、 K_1^* と K_0^* は共に一様ランダムに選ばれるため、 $\mathcal{A}$ の視点からは、ゲームがチャレンジビットに依らず同一となることから、 $\Pr[S_4 \wedge \overline{D_4}] = \frac{1}{2}\Pr[D_4]$ となるためである。

第五に、 $\mathcal{C}$ の秘匿性により、 $|\Pr[D_4] - \Pr[D_5]|$ が無視できることを示せる。ゲーム4とゲーム5の違いは、 tag^* の平文として c_{in}^* を用いるか 0^n を用いるかであり、さらに tag^* を作成するための乱数 r_1^* には一様乱数が用いられる。従って、チャレンジ平文として c_{in}^* と 0^n を用い、 $\mathcal{A}$ に対してゲーム4をシミュレートし、 $\mathcal{A}$ が危険クエリを行うときのみ1を出力する様な帰着アルゴリズムを考えれば、 $\mathcal{C}$ の秘匿性をに対するアドバンテージが $|\Pr[D_4] - \Pr[D_5]|$ となり、 $\mathcal{C}$ の秘匿性により無視できる。

第六に、 $\mathcal{T}$ のeCPA安全性により、 $|\Pr[D_5] - \Pr[D_6]|$ が無視できることを示せる。ゲーム5とゲーム6の違いは、 c^* の平文として c_{in}^* を用いるか 0^n を用いるかであり、さらに c^* を作成するための乱数 r_2^* には一様乱数が用いられる。従って、チャレンジタグとして $\text{tag}^*(0^n \text{ のコミットメント})$ 、チャレンジ平文として c_{in}^* と 0^n を用い、 $\mathcal{A}$ に対してゲーム5をシミュレートし、 $\mathcal{A}$ が危険クエリを行うときのみ1を出力する様な $\mathcal{T}$ のeCPA安全性に対する攻撃者(帰着アルゴリズム)を考えることができる。eCPA攻撃者は、穴あき秘密鍵 $\widehat{sk}_{\text{tag}^*} = \text{Punc}(sk, \text{tag}^*)$ も与えられるため、 $\mathcal{A}$ の復号クエリが危険クエリかどうかの判定、及び $\text{AltDecap}(\widehat{SK}_{\text{tag}^*}, C)$ の実行も問題無く行うことができる。この帰着アルゴリズムのeCPAアドバンテージは $|\Pr[D_5] - \Pr[D_6]|$ となり、 $\mathcal{T}$ のeCPA安全性により無視できる。

最後に、 Γ_{in} の予測不可能性により、 $\Pr[D_6]$ が無視できることを示せる。ゲーム6では、 C^* は c_{in}^* の情報を一切含まない。従って、 $\mathcal{A}$ がゲーム6で危険クエリを行う確率は、 $\mathcal{A}$ (に対しゲーム6をシミュレートする帰着アルゴリズム)が $\mathcal{T}$ の予測不可能性を破る確率で上限でき、 Γ_{in} の予測不可能性によりその確率は無視できる。

以上と不等式(1)から、 $\mathcal{A}$ のCCAアドバンテージは無視できることが分かる。□

4 比較と拡張

本節では、提案方式及び既存方式によってCCA安全な1ビットPKEのみからCCA安全なKEMを構成した場合の、公開鍵サイズと暗号文サイズを、構成要素の1ビットPKEのサイズを基に比較する。

以下ではCCA安全な1ビットPKEの公開鍵サイズと暗号文サイズをそれぞれ $|pk|$ と $|c|$ 、ターゲット衝突困難ハッシュの鍵サイズを $|hk|$ 、ワンタイム署名の検証鍵及び署名のサイズをそれぞれ $|vk|$ 及び $|\sigma|$ と書く。

既存方式 CCA安全な1ビットPKEからCCA安全なKEMを構成する既存の(著者らが知る限り)最も効率の良い方法は、[8]と[3]を組み合わせることにより得られる。[8]は、[6]の構成法においてハイブリッド暗号の考え方をを用いることで暗号文サイズの効率化を図ったものであり、構成要素として、CPA安全なPKE、1-CCA安全なKEM、1-CCA安全なSKE、DCCA安全なDKEM、OT安全なSKE、を用いたCCA安全なKEMの構成法を示した。これらの内、CPA安全なPKEとDCCA安全なKEMは、CCA安全な1ビットPKEの連結構成(とPRGの組み合わせ)により得られる。これらの公開鍵と暗号文サイズはそれぞれ $|pk|$ と $k|c|$ となる。また、著者らが知る限り、1-CCA安全なKEMをCCA安全な1ビットPKEから最も効率よく構成する方法は、DodisとFiore [3, Appendix C]によるCPA安全なPKEとワンタイム署名を用いるPKEの構成において、CCA安全な1ビットPKEによる連結構成を(平文空間が k ビットの)CPA安全なPKEとして用い、さらに k ビットの乱数 K を暗号化してKEMとして用いる方法である。この1-CCA安全なKEMは、いわゆるDDN-Lite構成(DDN構成[4]から非対話型ゼロ知識証明を除いた構成)と同等の公開鍵サイズ $2k|pk| + |hk|$ と暗号文サイズ $k^2|c| + |vk| + |\sigma|$ を持つ。さらに、1-CCA安全なSKEとOT安全なSKEは、暗号文サイズのオーバーヘッド無しで実現できる

[15]。(これら SKE は一方向関数のみから実現できるため、CCA 安全な 1 ビット PKE から実現できる。)

上記の各構成要素を [8] において用いた場合、構成された CCA 安全な KEM の各サイズは以下の通りとなる:

- 公開鍵サイズ: $(2k+2)|pk| + |hk|$
- 暗号文サイズ: $(k^2+2k)|c| + |vk| + |\sigma|$

提案方式 3 節で示した提案方式を用いて、CCA 安全な 1 ビット PKE から CCA 安全な KEM を構成する場合を考える。構成要素の PTBE として、[10] で示された構成法を用いる。(k ビットよりも長い値を暗号化する必要があるため、2.2 節で説明したハイブリッド暗号の方法を用いる。) [10] の構成の中で用いられる CPA 安全な PKE には、1 ビット PKE の連結構成を用いる。また、コミットメント方式として、2.3 節で説明した CPA 安全な PKE とターゲット衝突困難ハッシュ関数を用いた構成を考える。さらに、DCCA 安全な DKEM には、上記と同様、1 ビット PKE の連結構成 (と PRG) を用いる構成を考える。

上記の各構成要素を 3 で示した提案方式において用いた場合、構成された CCA 安全な KEM の各サイズは以下の通りとなる:

- 公開鍵サイズ: $(2k+2)|pk| + |hk|$
- 暗号文サイズ: $(k^2+k)|c| + k$

公開鍵サイズは既存方式と提案方式とで違いは無いが、暗号文サイズは提案方式の方が小さくなっていることが分かる。

拡張 2.3 節で説明した通り、提案方式では 2 メッセージ型のコミットメント方式を用いることができる。(例えば Naor [12] のコミットメント方式を 2.3 節で説明したターゲット衝突困難ハッシュと組み合わせる手法を用いることができる。) この場合、提案方式の公開鍵サイズは $(2k+1)|pk| + |ck| + |hk|$ となる。ただし、 $|ck|$ はコミットメント方式の受信者から送信者への最初のメッセージである。用いる 1 ビット PKE とコミットメント方が、 $|pk| > |ck|$ を満たす場合、提案方式の公開鍵サイズの方が小さくなる。

また、[10] の PTBE や Dodis と Fiore の 1-CCA 安全な PKE の構成 [3, Appendix C] では、Dowsley ら [5] の手法により暗号文サイズを半分にすることができる。従って、提案方式と既存方式共に、暗号文サイズにおける k^2 の項が $(1/2)k^2$ となる様な構成が可能である。

さらに、提案方式と既存方式共に、1 ビット PKE は必ずしも CCA 安全である必要はなく、

DCCA 安全かつ予測不可能性を満たす検出可能 PKE [6] であればよい。CCA 安全な 1 ビット PKE は DCCA 安全性かつ予測不可能性を満たす検出可能 PKE としてみなすことができるが、逆は必ずしも成り立たないため、後者の方が弱い仮定である。

参考文献

- [1] R. Cramer, G. Hanaoka, D. Hofheinz, H. Imai, E. Kiltz, R. Pass, A. Shelat, and V. Vaikuntanathan. Bounded CCA2-secure encryption. *ASIACRYPT 2007*, LNCS 4833, pp. 502–518, 2007.
- [2] R. Cramer and V. Shoup. Design and analysis of practical public-key encryption schemes secure against adaptive chosen ciphertext attack. *SIAM J. Computing*, 33(1):167–226, 2003.
- [3] Y. Dodis and D. Fiore. Interactive encryption and message authentication, 2013. <http://eprint.iacr.org/2013/817>. To appear in SCN 2014.
- [4] D. Dolev, C. Dwork, and M. Naor. Non-malleable cryptography. *STOC 1991*, pp. 542–552, 1991.
- [5] R. Dowsley, G. Hanaoka, H. Imai, and A.C.A. Nascimento. Reducing the ciphertext size of Dolev-Dwork-Naor like public key cryptosystems, 2009. <http://eprint.iacr.org/2009/271>.
- [6] S. Hohenberger, A. Lewko, and B. Waters. Detecting dangerous queries: A new approach for chosen ciphertext security. *EUROCRYPT 2012*, LNCS 7237, pp. 663–681, 2012.
- [7] P. MacKenzie, M.K. Reiter, and K. Yang. Alternatives to non-malleability: Definitions, constructions and applications. *TCC 2004*, LNCS 2951, pp. 171–190, 2004.
- [8] T. Matsuda and G. Hanaoka. Achieving chosen ciphertext security from detectable public key encryption efficiently via hybrid encryption. *IWSEC 2013*, LNCS 8231, pp. 226–243, 2013.
- [9] T. Matsuda and G. Hanaoka. Chosen ciphertext security via point obfuscation. *TCC 2014*, LNCS 8349, pp. 95–120, 2014.
- [10] T. Matsuda and G. Hanaoka. Chosen ciphertext security via UCE. *PKC 2014*, LNCS 8383, pp. 56–76, 2014.
- [11] S. Myers and A. Shelat. Bit encryption is complete. *FOCS 2009*, pp. 607–616, 2009.
- [12] M. Naor. Bit commitment using pseudorandomness. *J. of Cryptology*, 4(2):151–158, 1991.
- [13] M. Naor and M. Yung. Universal one-way hash functions and their cryptographic applications. *STOC 1989*, pp. 33–43, 1989.
- [14] M. Naor and M. Yung. Public-key cryptosystems provably secure against chosen ciphertext attacks. *STOC 1990*, pp. 427–437. ACM, 1990.
- [15] D.H. Phan and D. Pointcheval. About the security of ciphers (semantic security and pseudo-random permutations). *SAC 2004*, LNCS 3357, pp. 182–197, 2005.