PSEC-KEM の安全性評価

2002 年 11 月

株式会社 日立製作所

PSEC-KEM アルゴリズム評価報告書

2002 年 11 月

株式会社 日立製作所

目 次

記号

$a \mid b$: 整数 $a(\neq 0)$, b に対して a は b を割り切る

$a \bmod n$: 整数 a, n に対して a を n で割った剰余 (通常は $\{0, 1, \ldots, n-1\}$ の中からとる)

$\lceil a \rceil$: 実数 a 以上の最小整数

$\mathbb{F}_{q^m}$: q^m 個の元からなる有限体 (q は標数, m は自然数)

$x||y$: ビット列 x, y の連結 (concatenation)

$x \oplus y$: ビット列 x, y (長さは同じとする) のビット毎の排他的論理和

1 はじめに

本報告書は, 鍵カプセル化メカニズム PSEC-KEM についての評価結果をまとめたものである. 評価対象とした PSEC-KEM の文献は以下のものである.

1. PSEC-KEM 仕様書
2. PSEC-KEM 自己評価書

ただし, これらは NTT コミュニケーションズ株式会社より支給された (2002 年 9 月) ものである.

鍵カプセル化メカニズム (key encapsulation mechanism: KEM) は, 従来のハイブリッド暗号方式 (メッセージ本体は共通鍵暗号により暗号化し, その鍵を公開鍵暗号で暗号化, 配送する方式) における公開鍵技術部分を抜き出し, 概念化したもので, 同様にして共通鍵技術部分を抜き出した概念であるデータカプセル化メカニズム (data encapsulation mechanism: DEM) と組み合わせることでひとつの非対称暗号化方式となるものである.

この定式化, 概念整備は ISO/IEC JTC1/SC27 における暗号標準化作業などでなされている ([32], [10]) もので公開鍵暗号技術の新しい展開として注目されている.

このような動きの中で, PSEC-KEM は, CRYPTREC2000 応募暗号である PSEC-2([24], [25]) の鍵カプセル化メカニズム版として CRYPTREC2001 に応募された. PSEC-KEM は楕円曲線上での ElGamal 暗号方式 ([13], [20]) をもとにして構成された方式で, 楕円曲線上の Diffie-Hellman 計算問題の困難性を前提に, ランダムオラクルモデル上で "安全" (鍵カプセル化メカニズムとして適応的選択暗号文攻撃に対して強秘匿) であることが証明されている.

本報告書では, 2 章で, 鍵・データカプセル化メカニズムの定義や安全性, さらにそれらを組み合わせて得られるハイブリッド方式の安全性など一般的概念を説明し, 3 章で, これらに関する ISO などにおける動向を報告する. 4 章では, PSEC-KEM の方式の概要を述べ, 5 章で, PSEC-KEM が安全性の根拠とする 楕円曲線上の Diffie-Hellman 計算問題や離散対数問題に関する状況, さらに PSEC-KEM の安全性証明の検討結果などについて報告する. 6 章では他の楕円 ElGamal 暗号ベースの KEM 方式との安全性, 効率性比較を行い, 最後に 7 章で本報告の結論を述べる.

2　カプセル化メカニズム

公開鍵暗号は一般に共通鍵暗号に比べ処理速度が遅く, 大容量データの暗号化には不向きである. 一方, 共通鍵暗号は二者間でデータのやり取りを行う場合, 何らかの方法で暗号化鍵を共有しなければならない. これら両者の長短所をうまく融合した非対称暗号化方式がハイブリッド方式である. ハイブリッド方式では, 暗号化したいデータ本体は共通鍵暗号で暗号化し, そのデータ暗号化鍵を, 公開鍵暗号を用いて暗号化する. 暗号文はデータ本体を暗号化したものと, データ暗号化鍵を暗号化したものの対となる.

通常の公開鍵暗号を単純に用いてハイブリッド方式を構成することができる. すなわち, 共通鍵暗号の鍵 (データ暗号化鍵) として乱数を生成し, それを公開鍵暗号に平文として入力, 暗号化すればよい. しかし, より直接的に, ハイブリッドで要求される機能に特化した方式を用いる方が様々な面で効率的である.

この流れで考案された概念がカプセル化メカニズムである. 公開鍵技術部分を鍵カプセル化メカニズム (KEM : key encapsulation mechanism) といい, 共通鍵技術部分をデータカプセル化メカニズム (DEM : data encapsulation mechanism) という.

以下に KEM, DEM についてより詳しく解説する.

2.1　KEM の定義

鍵カプセル化メカニズムは次の 3 つのアルゴリズムからなる.

鍵生成アルゴリズム $KEM.KeyGen()$: 入力はなく, 公開鍵/秘密鍵の組 (PK, pk) を出力する.

暗号化アルゴリズム $KEM.Encrypt(PK, option)$: 公開鍵 PK を入力とし (さらに暗号化オプション $option$ を入力として持ってもよい), データ暗号化鍵/暗号文 の組 (K, C_0) を出力する.

復号化アルゴリズム $KEM.Decrypt(pk, C_0)$: 秘密鍵 pk と, 暗号文 C_0 を入力とし, データ暗号化鍵 K を出力する. 復号化アルゴリズムはある条件のとき **fail** してもよい.

鍵カプセル化メカニズムは, $KEM.Encrypt$, $KEM.Decrypt$ で出力するデータ暗号化鍵の長さ $KEM.KeyLen$ も指定する. また, 通常の公開鍵暗号と同様に, ある公開鍵 PK で暗号化された暗号文 (K, C_0) は, 対応する秘密鍵 pk により K に復号されねばならない. ただし, 鍵生成アルゴリズムにおいて, これが成り立たない鍵の組 (PK, pk) が生成される確率が無視できるほど小さければよいとする.

上記定義からわかるように, 鍵カプセル化メカニズムは基本的に入力が公開鍵のみで, 平文に相当する入力を持たない. これが通常の守秘目的公開鍵暗号との違いであり (図 1), 当然のことながら 鍵カプセル化メカニズム単体では守秘等の目的には用いることができない. あくまでも後述のデータカプセル化メカニズムと組み合わせ, ハイブリッド型にすることで初めて暗号として機能するものである.

2.2　KEM の安全性の定義

鍵カプセル化メカニズムは, 通常の公開鍵暗号とは異なるので安全性の概念も若干異なる. ここではその安全性の概念について説明する.

次のようなゲーム (適応的選択暗号文攻撃) を考える:

Stage 1 : 鍵生成アルゴリズムを走らせ, 公開鍵/秘密鍵の組を生成する. 攻撃者には公開鍵のみを与えるものとする.

Stage 2 : 攻撃者は**復号化オラクル**へ任意回の質問 (query) をする. 各質問は暗号文であり, 復号化オラクルにより復号され攻撃者に与えられる. もし復号が **fail** ならばその情報も攻撃者に与えられる. 攻撃者は復号化オラクルへの質問 (暗号文) をどのようにして作成しても構わない.

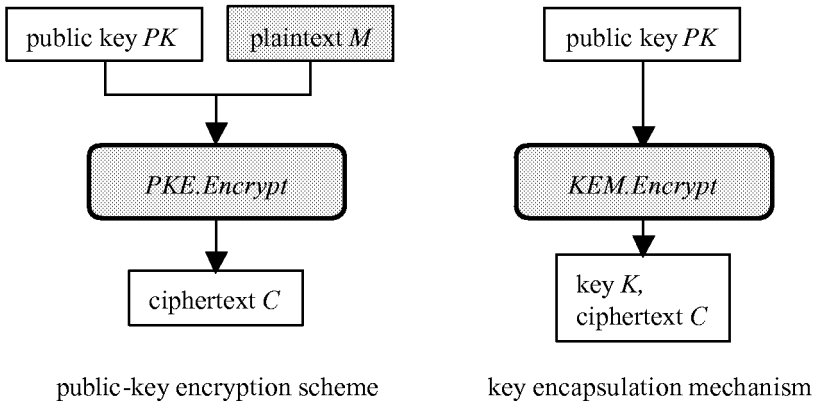

図 1: 公開鍵暗号と鍵カプセル化メカニズム: 暗号化

Stage 3 : 攻撃者は**暗号化オラクル**を呼び出す. 暗号化オラクルは次を実行する.

1. 暗号化アルゴリズムを走らせ, (K^*, C_0^*) を生成する.
2. 長さ $KEM.KeyLen$ の bit 列 $\tilde{K}$ をランダムに生成する.
3. ランダムに $b \in \{0,1\}$ を選ぶ.
4. $b = 0$ ならば (K^*, C_0^*) を出力し, そうでなければ $(\tilde{K}, C_0^*)$ を出力する.

Stage 4 : 攻撃者は復号化オラクルへ質問を続ける. ただしその質問は C_0^* とは異なるものとする.

Stage 5 : 攻撃者は $\hat{b} \in \{0,1\}$ を出力して終了する.

ある鍵カプセル化メカニズム KEM に対し, 上記攻撃における攻撃者 A の **advantage** を

$$Adv_{KEM}(A) = |\Pr[\hat{b} = b] - 1/2|$$

と定義する. KEM に対し, A が時間 t, 復号化オラクルへの質問が高々 q 回で動くアルゴリズムであるとき, A を $KEM[t,q]$-adversary と呼ぶ.

鍵カプセル化メカニズムが**安全**であるとは, 任意の攻撃者 (確率的多項式時間アルゴリズム) に対して advantage が無視できるほど小さいことをいう. advantage を, $Adv'_{KEM}(A) = |\Pr[\hat{b} = 1 \mid b = 0] - \Pr[\hat{b} = 1 \mid b = 1]|$ とすることもある. 容易にわかるように, $Adv'_{KEM}(A) = 2 \cdot Adv_{KEM}(A)$ が成り立つ.

最初に述べたように, 通常の意味の公開鍵暗号方式 PKE が与えられたとき, 外部で乱数を生成し, それを平文として PKE により暗号化することで KEM 方式を得る (図 2). このようにして作られる KEM を PKE_{KEM} と書くことにする.

PKE が公開鍵暗号として安全 (IND-CCA2[†]) であるならば, 容易にわかるように, PKE_{KEM} は KEM とし

[†]一般に, 非対称暗号が安全であることは次のように定義される ([32]).
次のようなゲームを考える:

Stage 1 : 鍵生成アルゴリズムを走らせ公開鍵/秘密鍵の組を生成する. 攻撃者には公開鍵のみを与えるものとする.
Stage 2 : 攻撃者は**復号化オラクル**へ任意回の質問 (query) をする. 各質問は暗号文であり, 復号化オラクルにより復号され攻撃者に与えられる. もし復号が **fail** ならばその情報も攻撃者に与えられる. 攻撃者は復号化オラクルへの質問 (暗号文) をどのようにして作成しても構わない.
Stage 3 : 攻撃者は (同じ長さの) 平文 M_0, M_1 を **暗号化オラクル**に与える. 暗号化オラクルは次を実行する.
1. ランダムに $b \in \{0,1\}$ を選び, 2. 暗号化アルゴリズムにより M_b を暗号化し, 暗号文 C_b を作成, 3. C_b を出力する.
Stage 4 : 攻撃者は復号化オラクルへ質問を続ける. ただしその質問は C_b とは異なるものとする.
Stage 5 : 攻撃者は $\hat{b} \in \{0,1\}$ を出力して終了する.

ある非対称暗号 AC に対し, 上記攻撃における攻撃者 A の **advantage** を $Adv_{AC}(A) = |\Pr[\hat{b} = b] - 1/2|$ と定義する. 非対称暗号が**安全** (適応的選択暗号文攻撃に対して強秘匿: **IND-CCA2**) であるとは, 任意の攻撃者 (確率的多項式時間アルゴリズム) に対して advantage が無視できるほど小さいことをいう. 頑強性 (non-malleability) の概念もあるが, 適応的選択暗号文攻撃のもとでは上記と等価な概念であることが証明されている ([2]).

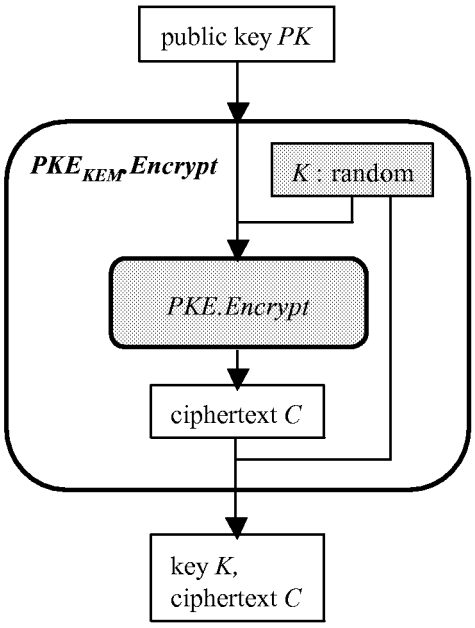

図 2: 公開鍵暗号による鍵カプセル化メカニズム: 暗号化

て上記意味で安全である. PKE_{KEM} への攻撃者 $\mathcal{A}_{KEM}$ が存在したとして, これを用いて以下のように (図 3 も参照) PKE への IND-CCA2 攻撃者 $\mathcal{A}_{PKE}$ を構成すればよい.

$\mathcal{A}_{PKE}$ は平文 M_0, M_1 を任意に選び, 暗号化オラクルにいずれかを暗号化してもらう. その暗号文 C_b $(b = 0, 1)$ がどちらの暗号文になっているかを当てなければならない. このとき, (M_0, C_b) を $\mathcal{A}_{KEM}$ に KEM 方式としての問題として入力すれば, $\mathcal{A}_{KEM}$ は仮定により, 無視できない確率で正しい平文 (KEM ではデータ暗号化鍵に対応する) とその暗号文の組であるか否かを当てることができるので, その出力をそのまま用いることで $\mathcal{A}_{PKE}$ も無視できない確率で (同じ確率で) 正解することができる. ただし, $\mathcal{A}_{KEM}$ からの復号化オラクルへの質問に対して, $\mathcal{A}_{PKE}$ は自身が利用できる復号化オラクルへそのまま質問すればよい (ランダムオラクルモデル上である場合も同様である).

2.3 DEM の定義

鍵カプセル化メカニズムを用いてハイブリッド暗号を構築するには, データカプセル化メカニズム (DEM : data encapsulation mechanism) の概念が必要である. データカプセル化メカニズム DEM は, 暗号化, 復号化アルゴリズムからなる. また鍵長 $DEM.KeyLen$ を指定する.

暗号化アルゴリズム $\boldsymbol{DEM.Encrypt}(K, L, M)$: データ暗号化鍵 K, ラベル L, および平文 M を入力とし暗号文 C_1 を出力する. ここで, M, L は任意長, K は長さ $DEM.KeyLen$ とする.
(M, L が実装時の限界長を超えているとき **fail** してもよい)

復号化アルゴリズム $\boldsymbol{DEM.Decrypt}(K, L, C_1)$: データ暗号化鍵 K, ラベル L, および暗号文 C_1 を入力とし平文 M を出力する. 復号化アルゴリズムはある条件のとき **fail** してもよい.

暗号化, 復号化アルゴリズムは決定性アルゴリズムでなければならない.
また, 任意の K, L, M に対し, ($DEM.Encrypt$ が **fail** しないならば)

$$DEM.Decrypt(K, L, DEM.Encrypt(K, L, M)) = M$$

が成り立たなければならない.

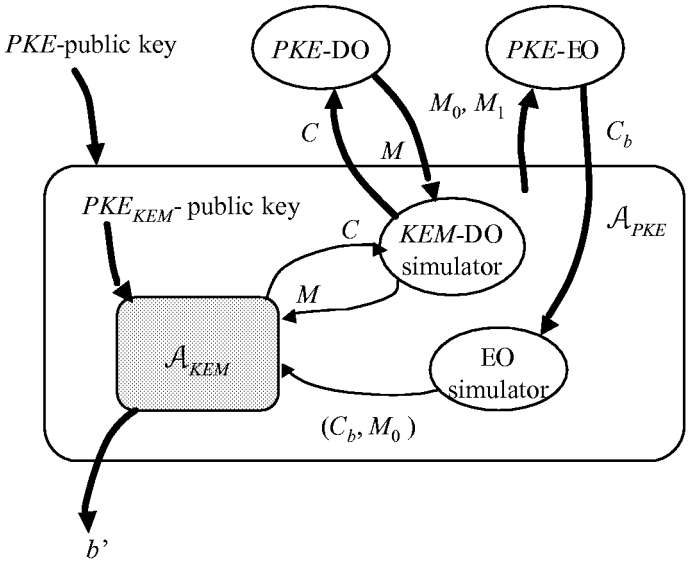

図 3: KEM の攻撃者を用いた PKE の攻撃者の構成

2.4 DEM の安全性の定義

データカプセル化メカニズムに対する攻撃のシナリオは次のとおり.

データカプセル化メカニズム *DEM* に対し, 攻撃者は同じ長さの平文 M_0, M_1, および, ラベル L^* を生成し, 暗号化オラクルに投げる. 暗号化オラクルはランダムにデータ暗号化鍵 K を生成し, また, ランダムに $b \in \{0,1\}$ を選択, 鍵 K, ラベル L^* で M_b を暗号化した暗号文 C_1^* を攻撃者に返す. 攻撃者は復号化オラクルに質問することができる. 質問はラベル/暗号文の組 (L, C_1) で, (L^*, C_1^*) と異なるものなら任意でよい. 復号化オラクルは鍵 K による復号結果を攻撃者に返す. 最後に攻撃者は $\hat{b}$ を出力する.

KEM と同様に, *DEM* に対する上記攻撃者 A の advantage を $Adv_{DEM}(A) = |\Pr[\hat{b} = b] - 1/2|$ と定める. さらに, 攻撃者 A が時間 t, 復号化オラクルへの質問が高々 q 回で, 復号化オラクルへの入力と, 暗号化オラクルからの出力の長さが高々 l, ラベルの長さが高々 l' で動くとき, A を *DEM*$[t, q, l, l']$-adversary と呼ぶ.

データカプセル化メカニズムが**安全**であるとは, 任意の攻撃者 (確率的多項式時間アルゴリズム) に対して advantage が無視できるほど小さいことをいう.

2.5 DEM に関する補足

ISO ドラフト ([32]) には, 共通鍵暗号と MAC アルゴリズムを用いて DEM を構成する方法が 3 つ (*DEM1*, *DEM2*, *DEM3*) 記載されている (ただし *DEM3* は MAC のみ). いずれも, 共通鍵暗号や MAC アルゴリズムが"安全"であれば, 構成された DEM が (DEM として) 安全であることも示されている (正確な証明は [10]).

共通鍵暗号としてはブロック暗号を (初期値固定の)CBC モードで用いるものと, one-time-pad 方式が記載されている (後述の, ハイブリッド暗号における共通鍵暗号に対する鍵 (データ暗号化鍵) は, ランダムなもので使い捨てであることから, 初期値固定の CBC モードであっても要求する安全性が確保できる). CBC モードでは, ブロック暗号が"pseudo-random permutation"であると仮定した場合, 求める安全性を達成することが証明されており, 安全な DEM の構成のための, 十分な安全性を持つと考える[†].

[†]KEM の安全性証明にランダムオラクル論を用い, それをもって安全と認める立場からは, ブロック暗号のモードに対するこのような証明も安全性の根拠として認められると考える. よって現実問題として PRP 的に見える具体的ブロック暗号 (AES, ...) を適応すれば安全な DEM が構成できると考える.

2.6 ハイブリッド暗号

ここでは KEM, DEM を組み合わせて 1 つの公開鍵暗号 $H\text{-}AC$ を構成する.
$KEM.KeyLen = DEM.KeyLen$ であるとする.

$H\text{-}AC$ の鍵生成は KEM と同じで, それを (PK, pk) とする.

暗号化アルゴリズム $H\text{-}AC.Encrypt(PK, L, M, option)$: 公開鍵 PK, ラベル L, 平文 M, および暗号化オプション $option$ を入力とし,
1. $(K, C_0) = KEM.Encrypt(PK, option)$,
2. $C_1 = DEM.Encrypt(K, L, M)$,
3. $C = C_0||C_1$,

を計算し, C を出力する.

復号化アルゴリズム $H\text{-}AC.Decrypt(pk, L, C)$: 秘密鍵 pk, ラベル L, 暗号文 C を入力とし,
1. C_0 が KEM の暗号文に適合する形であるよう $C = C_0||C_1$ と分割する[†]. このような分割ができないときは **fail** する.
2. $K = KEM.Decrypt(pk, C_0)$,
3. $M = DEM.Decrypt(K, L, C_1)$,

を計算し, M を出力する.

2.7 ハイブリッド暗号の安全性

$H\text{-}AC$ は機能として公開鍵暗号そのものであり, 安全性の概念もそれに従う. すなわち適応的選択暗号文攻撃に対して強秘匿であることが求められる. ただし, 平文長が任意, ラベルの存在の 2 点で通常の概念とは微妙に異なっている. 以下に $H\text{-}AC$ に対する攻撃モデルを簡単に説明する (3 ページ脚注参照. 詳細は [32], [10] を参照のこと).

攻撃者は復号化オラクルに能力の許す限り何度でも, ラベル/暗号文の組を質問し, その復号結果を得ることが許されている. 攻撃者は**同じ長さの**平文 M_0, M_1 と, ラベル L^* を暗号化オラクルに投げ, 暗号化オラクルはランダムに選択した $b \in \{0, 1\}$ に従って M_b をラベル L^* で暗号化, その暗号文 C^* を攻撃者に返す. 攻撃者はターゲットの (L^*, C^*) 以外のラベル/暗号文の組を復号化オラクルに質問し続け, 最後に $\hat{b} \in \{0, 1\}$ を出力する.

KEM, DEM と同様に, $H\text{-}AC$ に対する上記攻撃者 A の advantage を $Adv_{H\text{-}AC}(A) = |\Pr[\hat{b} = b] - 1/2|$ と定める. さらに, 攻撃者 A が時間 t, 復号化オラクルへの質問が高々 q 回で, 復号化オラクルへの入力と, 暗号化オラクルからの出力の長さが高々 l, ラベルの長さが高々 l' で動くアルゴリズムのとき, A を $H\text{-}AC[t, q, l, l']$-adversary と呼ぶ.

ハイブリッド暗号が**安全**であるとは, 任意の攻撃者 (確率的多項式時間アルゴリズム) に対して advantage が無視できるほど小さいことをいう.

2.6 節で構成した KEM, DEM によるハイブリッド暗号 $H\text{-}AC$ のに対し, KEM, DEM がそれぞれの意味で安全である場合, $H\text{-}AC$ も安全であることが示される. より正確に, 次が成り立つ:

任意の $H\text{-}AC[t, q, l, l']$-adversary A に対し, $KEM[t_1, q]$-adversary A_1 $(t_1 \approx t)$ と, $DEM[t_2, q, l, l']$-adversary A_2 $(t_2 \approx t)$ が存在して次が成り立つ.

$$Adv_{H\text{-}AC}(A) \leq 2 \cdot Adv_{KEM}(A_1) + Adv_{DEM}(A_2) \tag{1}$$

[†]厳密には KEM の出力に対し **prefix-freeness** などの性質を定義する必要があるが, ここでは簡単のため省略する. 詳細は [32] 参照.

式 (1) の証明の概要

ここでは簡単のためラベルのない場合を考える. 詳細は [10].

$\mathbf{G}_0$ をハイブリッド暗号の安全性における, もともとのゲームとし, *H-AC* への攻撃における暗号化オラクル内で生成される "hidden bit" を b, データ暗号化用鍵 (DEM 用鍵) を k^*, さらに暗号化オラクルの出力 (ターゲットの暗号文) を $C^* = C_0^*||C_1^*$ とする. $\hat{b}$ を攻撃者 A の出力とし, T_0 を *H-AC* 攻撃者 A が正解 ($\hat{b} = b$) を出す事象とする.

$\mathbf{G}_0$ を少し変形したゲーム $\mathbf{G}_1$ を次のように定義する. $\mathbf{G}_1$ では, 暗号化オラクルが呼ばれたとき, KEM の暗号化が終わった後, k^* を全くランダムな k^+ に置き換えて DEM の暗号化を行う. また, 出力 $C^+ = C_0^*||C_1^+$ を攻撃者に与えた後, $C = C_0||C_1$ が復号化オラクルに入力されたとき, $C_0 = C_0^*$ ならば KEM の復号化オラクルを用いず, 単に 暗号化オラクル内で作られた k^+ を用いるようにする. T_1 を A がゲーム $\mathbf{G}_1$ で正解する事象とする.

このとき, A の $\mathbf{G}_0$ での出力状況と, $\mathbf{G}_1$ での状況をみて KEM への攻撃を行う攻撃者を A_1 とするとき

$$|\Pr[T_1] - \Pr[T_0]| = Adv'_{KEM}(A_1)$$

である. 最後に, DEM への攻撃者 A_2 であって,

$$|\Pr[T_1] - 1/2| = Adv_{DEM}(A_2)$$

なるものが存在する. 実際, A はゲーム $\mathbf{G}_1$ においては単に *DEM* への攻撃をしていることになるので, A_2 は A と細部を除いて同じとみてよい. これら二つの式により, 求める不等式を得る[†].

[†]要するに, A はゲーム $\mathbf{G}_1$ において *DEM* への攻撃能力を発揮し, $\mathbf{G}_0$ と $\mathbf{G}_1$ の差において *KEM* への攻撃能力を発揮する.

3 ISOにおけるカプセル化メカニズム標準化の動向

ISO/IEC JTC1/SC27 では, 1999 年より暗号方式の標準化 (18033) を行っている. 18033 は, Part 1 (General), Part 2 (公開鍵), Part 3 (ブロック暗号), Part 4 (ストリーム暗号) の 4 つのパートから構成されている.

Part 2 (エディタ: V.Shoup) では, 通常の公開鍵暗号方式に加え, 任意長の平文を効率的に暗号化できるハイブリッド暗号にも注目し, ハイブリッド暗号での公開鍵暗号技術, 共通鍵暗号技術を切り分けた概念であるカプセル化メカニズムを提案, 標準化を進めている.

現在の Part 2 の WD(working draft [32]) では, ハイブリッド暗号における公開鍵技術部分を鍵カプセル化メカニズム (KEM), 共通鍵技術部分をデータカプセル化メカニズム (DEM) と名付け, それぞれの, 方式や安全性の定義, さらには, それらを組み合わせて安全なハイブリッド暗号を構成する方法等, これらの概念の一般的解説も記載されている.

PSEC-KEM は鍵カプセル化メカニズム (KEM) として part 2 に提案されている. Part2 には, PSEC-KEM の他に, 以下の表 3 に示す KEM の方式が提案されている (RSA-KEM については [19] も参照されたい[†]).

表 1: ISO に提案されている KEM 方式

	ベースとなる方式
PSEC-KEM	ElGamal
ACE-KEM	ElGamal
ECIES-KEM	ElGamal
RSA-KEM	RSA

Part 2 は, 2002 年 10 月に行われたワルシャワでの SC27 の国際会議において, WD から CD(committee draft) に進むことが承認されているが, 議論は流動的で, 表に示されている全ての方式が IS 化されるかどうかは現時点では不明である.

データカプセル化メカニズム DEM については, Part 2 の WD において概念および (ブラックボックス的) 構成法の概要は記述されているが, 具体的な DEM の方式については記載されていない. しかし, 現在のところ標準化方針は未定ながら, 別途 SC27 にて DEM の標準化が行われることがワルシャワ会議で決定されており, 今後, DEM 標準化は急速に行われると予想される.

[†]B. Kaliski によれば, RSA-KEM における RSA 関数部分を, 落とし戸付き一方向性関数に一般化した形にしても安全性の証明は可能である.

4 PSEC-KEM

ここでは鍵カプセル化メカニズム PSEC-KEM のアルゴリズムの概略を説明する. 有限体の元や, 楕円曲線上の点とビット列, オクテット列の相互変換などはある方法を選択, 固定しているものとし, 断わりなく同一視する. 詳細は仕様書を参照されたい.

4.1 鍵

楕円曲線パラメータ E とは 9 つの値の組 $(q, m, f(\beta), \boldsymbol{a}, \boldsymbol{b}, P, p, pLen, qmLen)$ のことをいう. ただし,

q : 素数, m : 正整数, $f(\beta)$: 有限体 $\mathbb{F}_q$ 上の m 次モニック既約多項式[†], $\boldsymbol{a}$, $\boldsymbol{b}$: $\mathbb{F}_{q^m}$ の要素, $P = (\boldsymbol{x}, \boldsymbol{y})$: 楕円曲線上の点, ただし $\boldsymbol{x}$, $\boldsymbol{y}$: $\mathbb{F}_{q^m}$ の要素で, $\boldsymbol{y}^2 = \boldsymbol{x}^3 + \boldsymbol{a}\boldsymbol{x} + \boldsymbol{b}$ $(q > 3)$, $\boldsymbol{y}^2 + \boldsymbol{x}\boldsymbol{y} = \boldsymbol{x}^3 + \boldsymbol{a}\boldsymbol{x}^2 + \boldsymbol{b}$ $(q = 2)$ を満たす. さらに, p : 素数, P の位数, $pLen$: $\lceil \log_2 p \rceil$ の値, $qmLen$: $\lceil \log_2 q^m \rceil$ の値

である.

PSEC 公開鍵 とは, 4 つ組み $(E, W, KDF, hLen)$ のことをいう. ただし, E : 楕円曲線パラメータ, W : 楕円曲線 (E) 上の点, KDF : 鍵導出関数の選択, $hLen$: 非負整数 である.

PSEC 秘密鍵 とは, 非負整数 s のことをいう

対応する PSEC 公開鍵, PSEC 秘密鍵においては, $0 \leq s < p$, $W = sP$ (楕円曲線 E 上の点としてのスカラー倍) の関係があり, これら公開, 秘密鍵は鍵生成プリミティブ (KGP-PSEC) により生成される.

4.2 暗号化処理

暗号化処理 (ES-PSEC-KEM-Encrypt) は入力 PSEC 公開鍵 PK に対し, (c_0, k) (オクテット列の組) を出力する. 処理は以下の手順で行われる:

1. エンコーディング処理 (EME-PSEC-KEM-A (図 4))

option の $keyLen$ が指定されているとする.

bit 長 $hLen$ の乱数 r を生成し, r から鍵導出関数 KDF により, bit 長 $pLen + 128 + keyLen$ の bit 列 H を生成, H を上位 $\{pLen + 128\}$-bit t と, 下位 $keyLen$-bit k (共有鍵) に分ける. t を整数とみなして mod$p$ での値を α (楕円演算用スカラー) とする.

2. 楕円演算処理 (EP-PSEC)

公開鍵である楕円曲線 E 上の点 P, W を, それぞれ α 倍する.

$$Q = \alpha W, \quad C_1 = \alpha P.$$

3. エンコーディング処理 (EME-PSEC-KEM-B (図 5))

楕円曲線上の点 C_1, Q から, 鍵導出関数 KDF により bit 長 $hLen$ の bit 列を生成, それと r の排他的論理和を c_2 とする.

4. 出力

C_1 と c_2 の連結 $c_0 = C_1 || c_2$, および 1 で求めた k を出力する.

[†]有限体の元の表現方法を固定したとき, 体の演算法を規定するための多項式. 言い換えると, 有限体 $\mathbf{F}_{q^m}$ の, $\mathbf{F}_q$ 上の定義多項式.

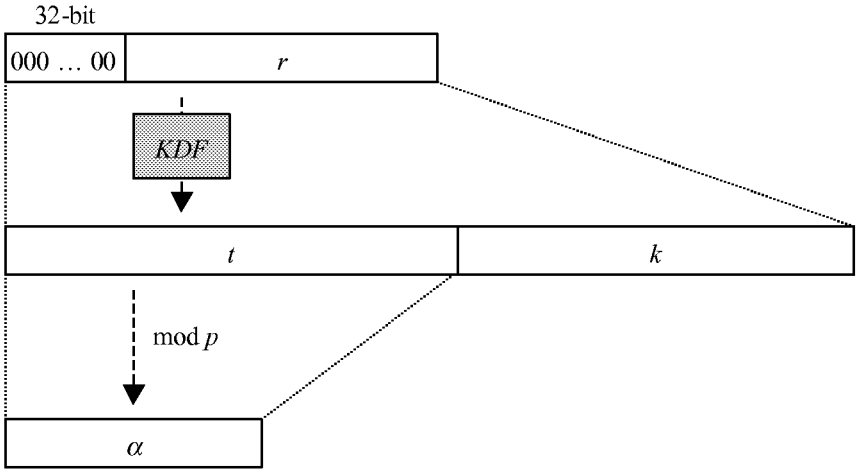

図 4: EME-PSEC-KEM-A

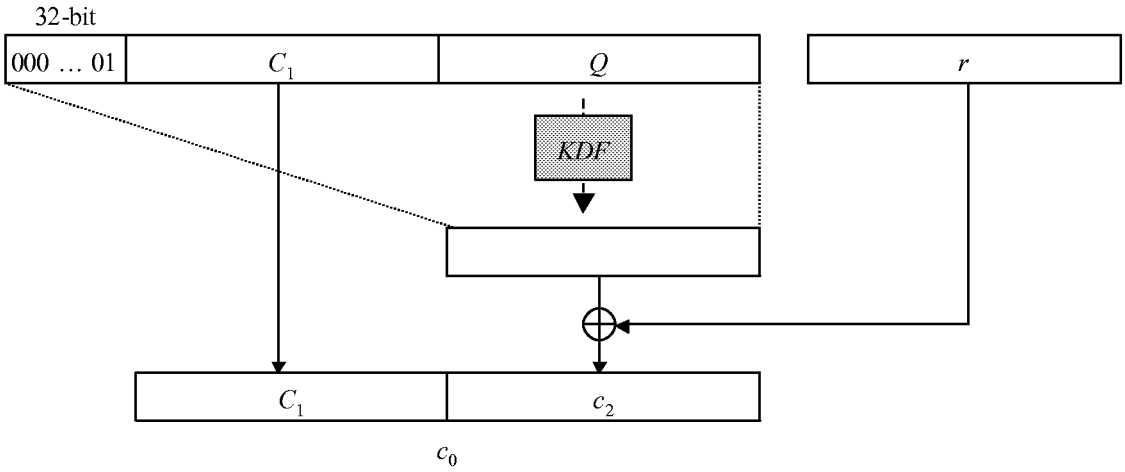

図 5: EME-PSEC-KEM-B

4.3 復号化処理

復号化処理 (ES-PSEC-KEM-DECRYPT) は入力 PSEC 公開鍵 PK, PSEC 秘密鍵 s, および 暗号文 c_0 に対し, k' (オクテット列の組) を出力する (エラー時, "invalid" を返すこともある). 処理は以下の手順で行われる:

1. デコーディング処理 (EME-PSEC-KEM-C)

c_0 の上位 $\{qmLen + 8\}$-bit から 楕円曲線 E 上の点 C'_1 を生成し, 下位 $hLen$-bit を c'_2 とおく.

2. 楕円演算処理 (DP-PSEC)

1 で求めた楕円曲線 E 上の点 C'_1 を 秘密鍵 s 倍した Q' を計算する.

$$Q' = sC'_1.$$

3. デコーディング処理 (EME-PSEC-KEM-D (図 6))

楕円曲線上の点 C'_1, Q' から, 鍵導出関数 KDF により bit 長 $hLen$ の bit 列を生成, それと c'_2 の排他的論理和を r' とする. r' から鍵導出関数 KDF により, bit 長 $pLen + 128 + keyLen$ の bit 列 h' を生成, h' を上位 $\{pLen + 128\}$-bit t' と, 下位 $keyLen$-bit k' (共有鍵) に分ける. t' を整数とみなして mod$p$ での値を α' とする.

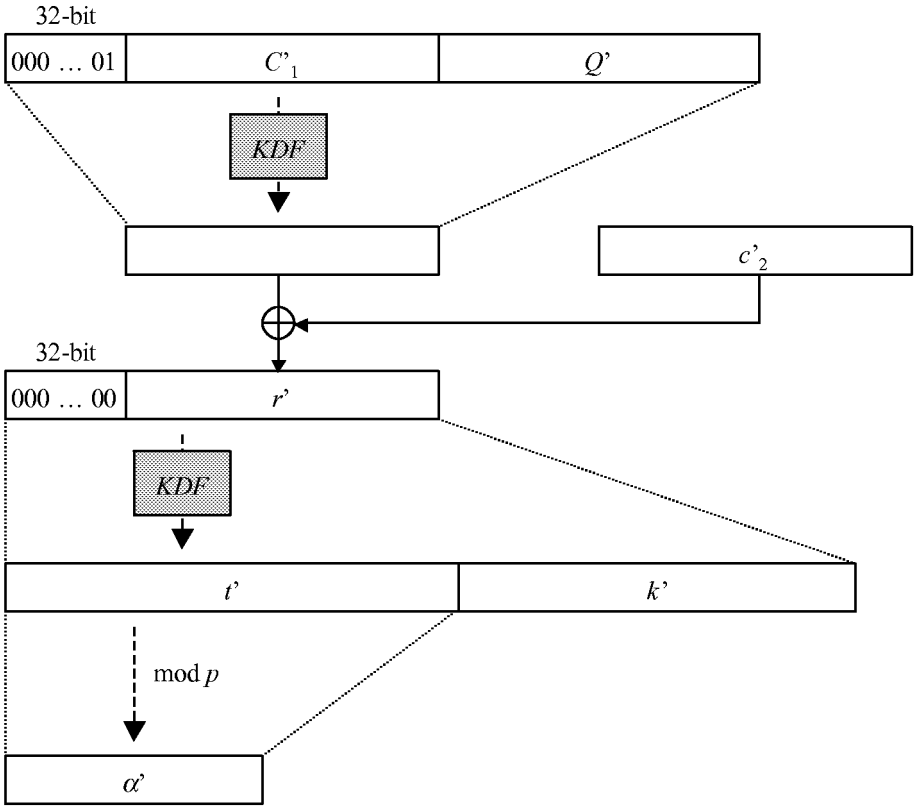

図 6: EME-PSEC-KEM-D

4. 検証, 出力 (DP-PSEC)

楕円曲線上の点として $C'_1 = \alpha' P$ が成り立てば k' (共有鍵) を出力する. そうでなければ "invalid" を返す.

5 PSEC-KEMの安全性

5.1 楕円曲線上のDL, CDH

ここではPSEC-KEMが安全性の根拠とする, 楕円曲線上のDiffie-Hellman計算問題 (CDH) について考察する.

Definition 5.1. (楕円曲線上の離散対数問題 : ECDL) $E = (q, m, f(\beta), \boldsymbol{a}, \boldsymbol{b}, P, p, pLen, qmLen)$ を楕円曲線パラメータ (*cf.* 4.1 節) とする. $x \in \{0, 1, \ldots, p-1\}$ をランダムに選んだとき, E と, xP から x を計算する問題を (E 上の) 離散対数問題 (DL : discrete logarithm problem) という.

Definition 5.2. (楕円曲線上の Diffie-Hellman 計算問題 : ECCDH) E を楕円曲線パラメータとする. $x, y \in \{0, 1, \ldots, p-1\}$ をランダムに選んだとき, E と, xP, yP から xyP を計算する問題を (E 上の)Diffie-Hellman 計算問題 (CDH : computational Diffie-Hellman problem) という.

明らかにCDHはDLに多項式時間帰着される. すなわち離散対数問題が解けるならばDiffie-Hellman計算問題も解くことができる. しかしその逆, つまり離散対数問題とDiffie-Hellman計算問題の計算量的等価性は一般には証明されていない

しかしながら, 部分的結果は報告されており, 例えば楕円曲線パラメータ E において, $p-1$, または $p+1$ が B-smooth (ただし, ある定数 c に対し $B = O((\ln p)^c)$) ならば E 上のDLとCDHは計算量的に等価である ([21]).

また, 素体上 (標数 q が smoothness condition を満たす) の楕円曲線上のDLが $L_{\frac{1}{2}}(q)$[†] の準指数時間で解けないならば, その楕円曲線上のCDHも $L_{\frac{1}{2}}(q)$ 時間で解けないことも示されている ([6]). ECDLは準指数時間では解けないと信じられており, ECCDHがECDLと同等の困難性を保持するであろうことの有力な根拠となると思われる.

以上のことより, 本報告書では, ECCDHの困難性より弱い仮定であるECDLの困難性が保証されることを"安全な楕円曲線"とみなすこととする.

一般的楕円曲線に対する攻撃

楕円曲線上の離散対数問題を解くアルゴリズムで準指数時間のものは知られておらず, 指数時間のものしかない. 現在知られている最も効率的なものはPollard ρ-法 ([28]) で, $\sqrt{\pi q}/2$ 回のステップを必要とする. 現在の計算機能力で (現実時間内で)ECDLを解くことが出来るのは楕円曲線パラメータにおける p : 素数部分群の位数が, およそ100数bit程度のものまでである. [12] では素体上の109-bit楕円曲線上のDLが (Pollard法で) 549日かけて解かれたことが報告されている[††].

この実態を考えると, 160-bit楕円曲線の場合, 単純計算で $2^{25} \approx 33,550,000$ 倍の計算量が必要で, 現在の計算機能力では5千万年以上かかることになる. 従って160-bit以上の楕円曲線を用いればPollard法に対しては十分安全であると考えることができる.

特殊楕円曲線に対する攻撃

一方で, 特殊な楕円曲線に対する攻撃もいくつか考案されており, これらの曲線を除かなければ, 離散対数問題の困難性を仮定することができない.

Weil/Tate-paring を用いてECDLを乗法群上のDLに持ち込む攻撃が, MOV攻撃 ([22]), FR攻撃 ([15]) など, いくつか知られている ([33] など). これらに対しては, 楕円曲線の有理点の群を乗法群に埋め込むことが困難, あるいは埋め込んでも乗法群の離散対数問題が困難であるように曲線を選ぶ必要がある ($p \mid (q^n - 1)$ なる n が十分大きければ ($n > 20$ など) よい).

また, anomalous 曲線も離散対数問題が易しく, 選択できない ([29]).

[†] $L_a(q) = \exp(\log^a q \log\log^{1-a} q)$

[††] 108-bit の 2 べき体上の楕円曲線も解かれている ([8]).

さらには, (真の) 拡大体上の楕円曲線に対し, Weil descent を利用した攻撃 ([16]) も知られており, $\mathbb{F}_{2^m}$, m : 合成数 などは避けるべきである.
(これら特殊楕円曲線への攻撃についての詳細は [11] を参照されたい.)

PSEC-KEM では, 楕円曲線の群のサイズ (p) は 160-bit を推奨しており, 楕円曲線パラメータの選択等は SECG([34]) を参照することになっている.
SECG に記載の楕円曲線は上記攻撃に対する安全性を十分考慮しておりこれらの楕円曲線パラメータを用いる限り, ECDL の困難性は (実用上) 保証されると考えてよい.

5.2 PSEC-KEM の安全性証明

ここでは PSEC-KEM 自己評価書に記載の安全性証明について検証する.

自己評価書では, 楕円曲線上の Diffie-Hellman 計算問題の計算量的困難性を仮定し, さらに, 暗号化処理 (4.2) の 1 における *KDF*, および復号化処理 (4.3) の 3 において 2 番目に用いられる *KDF* をランダムオラクル G とみなし, 暗号化処理の 3 における *KDF*, および復号化処理の 3 において最初に用いられる *KDF* を G とは独立なランダムオラクル H とみなした上で, PSEC-KEM が KEM として安全 (*cf.* 2.2 節) であることの証明を行っている (ランダムオラクルモデルでの証明).

より詳細に, 次の定理が示されている:

Theorem 5.3. (PSEC-KEM の安全性)[†] $\mathcal{A}$ を, 復号オラクル, ランダムオラクル G, H にそれぞれ q_D 回, q_G 回, q_H 回の質問をし, アドバンテージが ε で 実行時間が t であるような *PSEC-KEM*$(\mathcal{K},\mathcal{E},\mathcal{D})$ に対する IND-CCA2(適応的選択暗号文攻撃に対して強秘匿) における攻撃者とする. このとき成功確率が ε' で実行時間が t' の ($\mathcal{K}$ が出力するパラメータに関する) 楕円 Diffie-Hellman(計算) 問題の回答を含む (q_H+q_D) 個の値のリストを出力するようなアルゴリズムが存在して,

$$\varepsilon' \geq \frac{\varepsilon}{2(1+2^{-128})} - \frac{(q_G+3q_D)(1+2^{-128})}{p} - \frac{2q_D+q_G}{2^{hLen}},$$
$$t' \leq t + q_H \cdot (T + \mathcal{O}(1)).$$

ここで T は $\mathcal{K}$ に関する楕円曲線上の乗算 2 回を行う計算時間である.

定理中の $\mathcal{K}$ は PSEC-KEM における (ランダムオラクルモデル上での) 鍵生成アルゴリズム, $\mathcal{E}$, $\mathcal{D}$ はそれぞれ暗号化アルゴリズム, 復号化アルゴリズムである.

上記定理は, *PSEC-KEM*$[t, q_D, q_H, q_G]$-adversary $\mathcal{A}$ であって, $Adv_{PSEC\text{-}KEM}(A) = \varepsilon$ なるものが存在するならば, 上記を満たす成功確率 ε' の, *ECCDH*$[t']$-adversary が存在することを主張している.

PSEC-KEM とある DEM 方式と組み合わせ, ハイブリッド暗号を構成する場合, 一般に KEM に対して要求されている 2.2 節の意味の安全性が満たされていれば, 得られるハイブリッド暗号が非対称暗号として最強レベルの安全性を持つことになるので, PSEC-KEM に対し, 上記定理が成り立てば十分である.

PSEC-KEM の安全性は上記のようにランダムオラクルモデル上で証明されており, 厳密には, 現実での安全性を示しているわけではない. 実際, ある特殊な状況を考えることで, ランダムオラクルモデル上では安全であっても, いかなる実環境でも安全とはならないような方式も示されている ([7]). しかし, PSEC-KEM の安全性証明から, 実環境においては, PSEC-KEM が安全でないならば, ECCDH が解けるか, もしくは *KDF* についての (ある種の) 攻撃が可能ということになり, 後者の攻撃が実際には困難 (無論 ECCDH が困難であることも仮定して) である限り, PSEC-KEM は安全であると考えることができる.

[†] 自己評価書 定理 3.1(7page)

このような考察から, 本報告では, ランダムオラクルモデル上での安全性証明を, 実環境での安全性を強力に示唆する論拠とみなし, 上記定理の成立をもって安全であると結論するものとする.

以上のことを考慮に入れ, PSEC-KEM 自己評価書記載の, 上記定理の証明の検証を行った. その結果, 証明に問題はなく, 主張が正しく示されていることを確認した. 従って, PSEC-KEM は, KEM として安全であることが確認された.

なお, security reduction も **tight** (*i.e.* $\varepsilon \approx \varepsilon'$) であり, 推奨パラメータ長の楕円曲線 (160 bit 以上), $hLen$(128 bit 以上) を使用する限り, 安全な KEM であると言える[†]. 従って安全なデータカプセル化メカニズムと組み合わせることにより, 安全なハイブリッド暗号を得ることができる.

参考までに, 以下に証明の概略 (成功確率の評価のみ) を記載する.

証明の概略

$\mathcal{A}$ を上記の $PSEC\text{-}KEM(\mathcal{K},\mathcal{E},\mathcal{D})$ への攻撃者 (*cf.* 2.2 節) とする. これを用いて ECCDH を解く帰着アルゴリズム $\mathcal{B}$ を以下のように構成する:

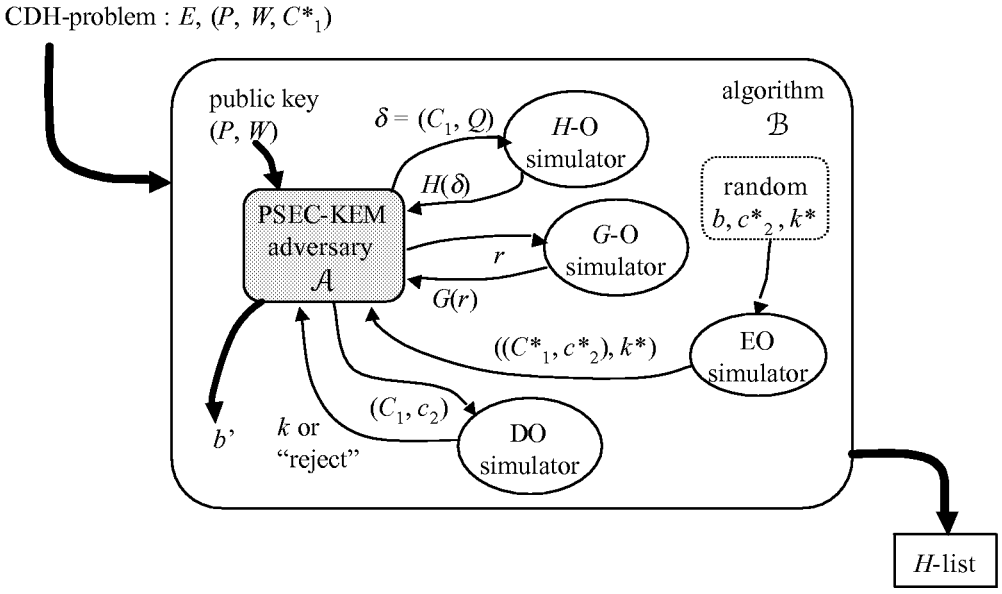

図 7: アルゴリズム $\mathcal{B}$

$\mathcal{B}$ の動作

$\mathcal{B}$ には楕円曲線 E 上の 3 点 P, W, C_1^* ($W, C_1^* \in \{P$ が生成する部分群 $\}$) が入力され, $\mathcal{B}$ は点 Q^* で $\log_{C_1^*} Q^* = \log_P W$ なるものを含むあるリストを出力する.

$\mathcal{B}$ はまずランダム bit b, およびランダムな bit 列 c_2^* (長さは H の出力長と等しい), k (長さは $keyLen$ に等しい) を生成する. 次に $\mathcal{A}$ を動かし, $\mathcal{A}$ からの H-query, G-query, および復号化オラクルへの query に答えながら (答え方は後述), $\mathcal{A}$ からの暗号化オラクルへの要求が来たら $\mathcal{A}$ に問題として $((C_1^*, c_2^*), k)$ を返す. 再び query の回答を, 行い, $\mathcal{A}$ が終了 (ある bit を出力する) したら $\mathcal{B}$ は H への query の中で C_1-part が C_1^* と一致しているもののリストを出力して終了する[††].

$\mathcal{A}$ からの query に対し, $\mathcal{B}$ は以下の動作をする:

[†]例えば $q_G = 2^{60}$, $q_D = 2^{40}$ と置いても 定理中の成功確率の式における右辺第 2 項以降は無視できて $\varepsilon' \approx \varepsilon/2$ であり, tight であると言える.

[††]CDH のランダム帰着性により, この出力から効率的に ECCDH の解答を導くアルゴリズムが知られている ([31], [32]).

H, G-オラクルのシミュレーション

H, G-query に対し, 初めて聞かれた質問に対しては (正しい長さの) ランダムな値を返す. query とそのランダムな回答をリストに記録する. 過去に聞かれた質問に対しては, その対応する回答を返せばよい.

ただし, H-query $\delta = (C_1, Q)$ において, $C_1 = C_1^*$ であるときは (上記に従い $H(\delta)$ を返した後,) $r = c_2^* \oplus H(\delta)$ が query として G-リストにあるか否かをチェックする. ある場合はそれでよいが, ない場合, r を G-リストに加え, $G(r)$ を以下のように定める:

$G(r)$ の上位 t-part をランダムに定め, $\alpha = t \bmod p$, αP を計算する.

もし $C_1^* = \alpha P$ であれば $Q^* = \alpha W$ (これは CDH の解答である) を計算し, $Q = Q^*$ ならば $G(r)$ の下位 k-part を, $b = 0$ のとき最初に選んだ k, $b = 1$ のときランダムな値にセットする. このとき, $((C_1^*, c_2^*), k)$ は対応する H, G オラクルの出力と矛盾のない, 正しい"問題"となっていることに注意.

$C_1^* \neq \alpha P$ のとき (ほとんどはこの場合となる) は $G(r)$ の k-part をランダムに定める.

最後に $r, G(r)$ を G-リストに加える.

復号化オラクルのシミュレーション

復号化オラクルへの query(C_1, c_2) に対しては, C_1-part が一致する H-リスト内の対を探し, なければ reject し, あるならば, そのような対全てに対して $r = c_2 \oplus H(\delta)$ を計算, r が G-query にあるか否かをチェックする. ない場合, (H-リストの上記のような対 1 つに対する)r に対し, $G(r)$ をランダムに選び, これらを G-リストに加える.

$G(r)$ の上位 t-part から (mod$p$ して) α を求め, $C = \alpha P$, および $Q = \alpha W$ が成り立つならば, $G(r)$ の下位 k-part を復号結果として $\mathcal{A}$ に返し, いずれかが成り立たなければ reject する.

復号化オラクルのシミュレーションでは, $\mathcal{A}$ が対応する H, G-query を事前に行っている場合には, 常に正しい復号シミュレーションができていることに注意する. よって, 復号化シミュレーションが失敗するのは $\mathcal{A}$ が H, G-query をすることなく, 正しい暗号文を作成できた場合である.

成功確率の評価

q_D 回の復号化オラクルへの query に対し, シミュレーションが 1 回以上の失敗 (間違った復号結果を出してしまう) する事象を **Fail** とするとき, Pr[**Fail**] を以下のように評価する.

$\mathcal{A}$ からの復号化オラクル query(C_1, c_2) に失敗したとする. $C_1 = C_1^*$ であるとき, ゲームのルールから, $c_2 \neq c_2^*$ である. $\mathcal{B}$ はこれを reject しているのだが, 失敗しているということは (C_1^*, c_2) は正しい暗号文であり, Q-part は等しい: $Q = Q^*$. ゆえに ($c_2 \neq c_2$ より) $r \neq r^*$ であるが, α-part ($\log_P C_1^* = \log_W Q^*$) は等しくなければならない. α-part は $G(r)$, $G(r^*)$ の上位 (t-part) を mod$p$ したものであり, これが等しくなる確率は (バイアスを考慮すると) $(1 + 2^{-128})/p$ となる. 複合化オラクルへの query は高々 q_D 回であり, また G オラクルに高々 q_G 回確認できるので, この場合 ($C_1 = C_1^*$ の場合), 失敗確率は高々

$$\frac{(q_G + q_D)(1 + 2^{-128})}{p}$$

となる.

次に $C_1 \neq C_1^*$ の場合, 対応する H, G-query のいずれかを行っていないのだが, まず, G-query を行っていないとする. しかし $\mathcal{A}$ は正しい暗号文を聞いてきているのだから, $G(r)$ の上位 (t-part) を mod$p$ したものが $\log_P C_1$ と一致していて, その確率は上と同様にして高々$(1 + 2^{-128})/p$, よってこの場合の失敗確率は高々

$$\frac{q_D(1 + 2^{-128})}{p}$$

となる.

最後に G-query は行うが, H-query を行わない場合, query が正しい暗号文であるのは $r' := c_2 \oplus H(C_1, Q)$ が r と一致するか (この確率は $1/2^{hLen}$), または $G(r')$ の t-part mod$p$ が $\log_P C_1$ と一致するか (この確率

は上と同様 $(1+2^{-128})/p)$ である. 従ってこの場合の失敗確率は高々

$$q_D\left(\frac{1}{2^{hLen}}+\frac{1+2^{-128}}{p}\right)$$

である. 以上のことにより

$$\Pr[\mathbf{Fail}] \leq \frac{(q_G+3q_D)(1+2^{-128})}{p}+\frac{q_D}{2^{hLen}} \tag{2}$$

を得る.

次に, $\mathcal{B}$ の目的である, (C_1^*, Q^*) が $\mathcal{A}$ からの H-query に含まれている確率を評価する. この事象を **AskH** とする.

また, r^* が G に質問され, その回答から得られる α-part が $C_1^* \neq \alpha P$ となってしまうか, もしくは k-part が $(b=0$ のとき) 出題の k と異なっている事象を **GBad** とする.

以下で, $\varepsilon' = \Pr[\mathbf{Fail}]$ が主張の不等式を満たすことを示す. 事象 $\mathbf{GBad} \vee \mathbf{Fail}$ に関して分けて考察する.

$$\begin{aligned}
\Pr[\mathbf{AskH} \wedge (\mathbf{GBad} \vee \mathbf{Fail})] &= \Pr[\mathbf{GBad} \vee \mathbf{Fail}] - \Pr[\neg\mathbf{AskH} \wedge (\mathbf{GBad} \vee \mathbf{Fail})] \\
&\geq \Pr[\mathbf{GBad} \vee \mathbf{Fail}] - \Pr[\neg\mathbf{AskH} \wedge \mathbf{GBad}] - \Pr[\neg\mathbf{AskH} \wedge \mathbf{Fail}] \\
&\geq \Pr[\mathbf{GBad} \vee \mathbf{Fail}] - \Pr[\mathbf{GBad}|\neg\mathbf{AskH}] - \Pr[\mathbf{Fail}]
\end{aligned}$$

であり, $\neg\mathbf{AskH}$ のとき, 値 $H(C_1^*, Q^*)$ は予測不可能で, よって r^* も予測不可能, 従って, r^* が G に聞かれる確率は $1/2^{hLen}$. ゆえに

$$\Pr[\mathbf{GBad}|\neg\mathbf{AskH}] \leq (q_D+q_G)/2^{hLen}$$

となる. よって式 (2) とあわせて

$$\Pr[\mathbf{AskH} \wedge (\mathbf{GBad} \vee \mathbf{Fail})] \geq \Pr[\mathbf{GBad} \vee \mathbf{Fail}] - \frac{(q_G+3q_D)(1+2^{-128})}{p} - \frac{2q_D+q_G}{2^{hLen}} \tag{3}$$

を得る.

一方, 否定部分は

$$\begin{aligned}
&\Pr[\mathbf{AskH} \wedge \neg(\mathbf{GBad} \vee \mathbf{Fail})] \\
&\quad = \Pr[\neg(\mathbf{GBad} \vee \mathbf{Fail})] \cdot \Pr[\mathbf{AskH}|\neg(\mathbf{GBad} \vee \mathbf{Fail})] \\
&\quad \geq \Pr[\neg(\mathbf{GBad} \vee \mathbf{Fail})] \cdot \Pr[(\mathcal{A}=b) \wedge \mathbf{AskH}|\neg(\mathbf{GBad} \vee \mathbf{Fail})] \\
&\quad \geq \Pr[\neg(\mathbf{GBad} \vee \mathbf{Fail})] \cdot (\Pr[\mathcal{A}=b|\neg(\mathbf{GBad} \vee \mathbf{Fail})] - \Pr[(\mathcal{A}=b) \wedge \neg\mathbf{AskH}|\neg(\mathbf{GBad} \vee \mathbf{Fail})].
\end{aligned}$$

ここでも $\neg\mathbf{AskH}$ のとき, 値 $H(C_1^*, Q^*)$ は予測不可能で, よって r^* も予測不可能, ゆえに $\mathcal{A}$ が正解する確率は $1/2$ となるので (このことは事象 $\neg(\mathbf{GBad} \vee \mathbf{Fail})$ とは独立であることも注意),

$$\Pr[(\mathcal{A}=b) \wedge \neg\mathbf{AskH}|\neg(\mathbf{GBad} \vee \mathbf{Fail})] \leq \frac{1}{2}$$

となる. また仮定から (バイアスも考慮すると)

$$\frac{\varepsilon}{2(1+2^{-128})}+\frac{1}{2} \leq \Pr[\mathcal{A}=b] \leq \Pr[\mathcal{A}=b|\neg(\mathbf{GBad} \vee \mathbf{Fail})] \cdot \Pr[\neg(\mathbf{GBad} \vee \mathbf{Fail})] + \Pr[\mathbf{GBad} \vee \mathbf{Fail}].$$

以上により,

$$\Pr[\mathbf{AskH} \wedge \neg(\mathbf{GBad} \vee \mathbf{Fail})] \geq \frac{\varepsilon/(1+2^{-128}) - \Pr[\mathbf{GBad} \vee \mathbf{Fail}]}{2} \tag{4}$$

となる. 式 (3), (4), および $\Pr[\mathbf{GBad} \vee \mathbf{Fail}] \geq 0$ から求める評価式を得る.

6 他方式との比較

ここでは, PSEC-KEM と, 他の楕円 ElGamal ベースの KEM 方式との比較 (安全性, 効率性等) を行う. 比較対象は, 代表的方式であり, ISO にも提案された (*cf.* 3 章) ACE-KEM, ECIES-KEM とする.

ACE-KEM は Cramer-Shoup 公開鍵暗号 ([9]) の KEM 版であり, ECICS-KEM は ECIES 公開鍵暗号 ([1], [34]) の KEM 版である. それぞれの詳細仕様は [32] に掲載されている. ここでは比較表のみを記載し, 比較結果を簡単に述べるにとどめる.

6.1 安全性比較

いずれの方式も KEM としての (2.2 節の意味での) 安全性は証明されているが, 証明のための仮定が異なっている (表 6.1 参照).

表 2: 安全性証明のための仮定

	数論仮定	関数仮定
PSEC-KEM	EC CDH	random
ACE-KEM	EC DDH	UOWH
ECIES-KEM	EC-GapCDH	random

ここで, ECDDH は (楕円曲線上の)Dillie-Hellman 決定問題を表し, EC-GapCDH は (楕円曲線上の)Gap-Diffie-Hellman(計算) 問題を表す.

詳細は省略するが, ECDDH は楕円曲線上の点の 4 つ組 (P, W, C, Q) が与えられたとき, Q が (P, W, C) に対する Diffie-Hellman 計算問題の解であるか否かを判定する問題であり, EC-GapCDH は, ECDDH の答えを返すオラクルを用いて, ECCDH を計算する問題である ([32], [27]).

また関数仮定における UOWH は universal one-way hash 関数 ([23]) を意味する. UOWH は一方向性関数が存在すれば構成でき, 数論仮定のもとで一方向性関数が構成できるので, UOWH は現実的仮定のもとで (効率性を問わなければ) 構成可能である.

表 6.1 では, 各数論問題が困難であること, および各方式におけるハッシュ関数部分において各関数仮定が成り立つことを仮定すれば, KEM としての安全性が証明可能であることを表している.

security reduction は, PSEC-KEM, ECIES-KEM ともほぼ最適である. それに比べ, ACE-KEM ではやや劣るものの, 特に問題ない.

ACE-KEM は, 関数仮定において, (理想的) ランダム関数に置き換えるランダムオラクル論は不要であり, 実環境に近い仮定において証明できる (スタンダードモデル上で証明可能) 方式で, 現実的安全性を確約することができる. ただし, 実際には UOW が何らかの意味で証明されたハッシュ関数を用いるわけではなく, 他方式同様, SHA-1([14]) 等の実用ハッシュ関数から構成したものを用いる. 従って, ACE-KEM の主張は, 他方式に比べ, ハッシュ関数の安全性への依存度がより低いことであると考えられる.

一方, ACE-KEM の数論仮定は ECDDH が困難であることであり, これは ECCDH より強い仮定である. これが成り立つには, ECCDH の場合以上に楕円曲線選択において注意を要する. 実際, 楕円曲線の群上に計算可能な (非自明な) 双線形写像が存在しさえすれば DDH 問題を解くことができ (値域での DDH や DL が計算可能である必要はない. また双線形写像の像の一部が計算できるだけでもよい), このような双線形写像は MOV 攻撃や FR 攻撃の対象より広範囲の楕円曲線において, Weil/Tate-paring を用いて構成可能である ([18], [5]). 今後, ECDDH の研究動向に注目する必要がある.

ECIES-KEM の数論仮定である EC-GapCDH([27]) は, 上記のような ECDDH の弱い楕円曲線 ([18], [5]) を想定した問題で, 明らかに ECCDH より強い仮定である[†]. 現在のところ, EC-GapCDH への攻撃で特筆すべきものはないが, 提出されて間もない問題であることから, 今後何らかの進展があることも否定できない.

6.2 効率性比較

表 6.2 は各方式の暗復号化効率をまとめたものである. 各暗復号化処理において, 楕円曲線上のスカラー倍算が大部分を占め, 他の処理は無視出来る程度であることから, 表ではスカラー倍算の回数を示している.

表 3: 楕円曲線上のスカラー倍算の回数

	暗号化	復号化
PSEC-KEM	2	2
ACE-KEM	5	3
ECIES-KEM	2	1

他方式に比べ, 明らかに ACE-KEM は処理効率が悪い. PSEC-KEM は復号が ECIES-KEM にやや劣るものの, ほぼ同等の効率性を持つとみてよい.

以上の比較を総合すると, PSEC-KEM の数論仮定は他に比べて信頼性の高い ECCDH であり, 処理効率性も十分よく, ランダムオラクルモデル上の安全性証明を認める立場においては, PSEC-KEM が最も優れた方式であるということができる.

[†]ECCDH が解けるならば, ECDDH オラクルなど必要なく EC-GapCDH が解けるのだから自明である.

7　結論

本報告書では, カプセル化メカニズムの概要を報告し, 鍵カプセル化メカニズムである PSEC-KEM の評価を行った.

まず, 鍵・データカプセル化メカニズムの定義や安全性, さらにそれらを組み合わせて得られるハイブリッド方式の安全性などを説明し, これらの概念が安全性に問題なく使用可能であることを確認した. またこれらの方式が ISO で急速に標準化されつつあることを報告した.

次に, 鍵カプセル化メカニズムである PSEC-KEM の概要を述べ, PSEC-KEM が安全性の根拠とする, 楕円曲線上の Diffie-Hellman 計算問題や離散対数問題に関する状況, さらに PSEC-KEM の安全性証明の検討などを行った. その結果, PSEC-KEM は推奨されているパラメータ長 (楕円曲線のサイズ, ハッシュ関数の出力値など) を用いる限り, KEM として安全であることを確認した. 従って "安全" な DEM と組み合わせることで公開鍵 (非対称) 暗号として最強レベルの IND-CCA2 (=NM-CCA2 [2]) を達成したハイブリッド暗号が構成できることになる.

さらに, 他の楕円 ElGamal ベースである鍵カプセル化メカニズムとの比較を行い, PSEC-KEM は, 他方式に対し優れた安全性 (信頼性), 効率性を持つことを述べた.

以上から PSEC-KEM は鍵カプセル化メカニズムとして優れた方式であることが結論付けられるが, 機能としては, あくまで鍵カプセル化のみを行う方式であり単体では用いることができない. 使用可能にするにはデータカプセル化メカニズムが必要である. このようなカプセル化メカニズムの概念は公開鍵暗号技術を有効に用いるために極めて重要と考えられ, 国内電子政府などでも必須の技術であると考える. カプセル化メカニズム関係の標準化, あるいは推奨方式の選択など, 早急に対応する必要があると考える.

最新動向の収得, 活用なくしては IT 産業の隆盛はありえず, 欧米, アジアの諸外国は これらの動きを注視し, 機敏に対応することで IT の諸分野における世界最先端の地位を獲得, 維持すべく鎬を削っている.

暗号界もますます流動的で, 新概念や強力な攻撃法が盛んに提案され続けており, ISO など, 国際的標準化活動はこれらに極めて敏感で, 対応も素早い.

一方, 我が国の暗号評価を行う本プロジェクトにおいては, 応募された暗号を評価する比較的長期の間, 技術仕様の修正・改良を一切認めないまま時間をかけて評価作業を行うという従来の運用では, 上記世界の動きの速さについていけない恐れがある. むしろ, 世界動向をいち早く取り入れ, より優れた技術を有効に活用するため, 本質的価値を優先した選定, 評価基準の検討や, それに沿った再評価, 追加・削除の仕組み整備など, 柔軟かつ迅速な対応を可能とする体制作りが望ましい.

さらには世界の動きに同調するのみでなく, 世界を動かしリードする技術の育成が重要であり, 本プロジェクトがこのような研究・開発を支援し, さらなる国益を念頭においた方針をとることが望ましいと思われる.

参考文献

[1] M. Abdalla, M. Bellare and P. Rogaway, The oracle Diffie-Hellman assumption and an analysis of DHIES, *Topics in Cryptology – CT-RSA 2001*, LNCS **2020** (2001), Springer-Verlag, 143-158.

[2] M. Bellare, A. Desai, D. Pointcheval and P. Rogaway, Relations Among Notions of Security for Public-Key Encryption Schemes, *Advances in Cryptology – CRYPTO '98*, LNCS **1462** (1998), Springer-Verlag, 26-45.

[3] M. Bellare and P. Rogaway, Random Oracles are Practical: A Paradigm for Designing Efficient Protocols, *Proceedings of the 1st ACM Conference on Computer and Communications Security*, ACM Press (1993), 62-73. Available at `http://www.cs.ucdavis.edu/ rogaway/papers/index.html`

[4] M. Bellare and P. Rogaway, Optimal Asymmetric Encryption - How to encrypt with RSA, *Advances in Cryptology – Eurocrypt '94*, LNCS **950** (1995), Springer-Verlag, 92-111.

[5] D. Boneh and M. Franklin, Identity-based encryption from the Weil pairing, *Advances in Cryptology – CRYPTO 2001*, LNCS **2139** (2001), Springer-Verlag, 213-229.

[6] D. Boneh and R. Lipton, Algorithms for black-box fields and their application to cryptography, *Advances in Cryptology – Eurocrypt '96*, LNCS **1109** (1996), Springer-Verlag, 283-297.

[7] R. Canetti, O. Goldreich and S. Halevi, The random oracle methodology, revisited, *Proceedings of the 30th Annual ACM Symposium on the Theory of Computing*, 1998, 209-218.

[8] Certicom ECC Challenge, 2002. Available at `http://www.certicom.com/research/news/ecc2k108.html`.

[9] R. Cramer and V. Shoup, A practical public key cryptosystem provably secure against adaptive chosen ciphertext attack, *Advances in Cryptology – Crypto'98*, LNCS **1462** (1998), Springer-Verlag, 13-25.

[10] R. Cramer and V. Shoup, Design and Analysis of Practical Public-Key Encryption Schemes Secure against Adaptive Chosen Ciphertext Attack, 2001. Available at `http://shoup.net/papers/` or `http://eprint.iacr.org/`

[11] CRYPTREC Report, Evaluation of security level of cryptography: The elliptic curve discrete logarithm problem, Dec. 14, 2001.

[12] The ECCp-109 challenge, 2002. Available at `http://www.nd.edu/ cmonico/eccp109/`

[13] T. ElGamal, A public key cryptosystem and a signature scheme based on discrete logarithms, *IEEE Transactions on Information Theory*, **31** (1985), 469-472.

[14] FIPS PUB 180-1, Secure Hash Standard (SHS), U.S. Department of Commerce / National Institute of Standards and Technology, April 17, 1995.

[15] G. Frey and H. G. Rück, A remark concerning m-divisibility and the discrete logarithm in the divisor class group of curves, Math. Comp. **62** (1991), 965-874.

[16] S. Galbraith and N. Smart, A Cryptographic Application of Weil Descent, HP Labs Tech. Report, HPL-1999-70.

[17] P. Gaudry, F. Hess and N. Smart, Constructive and destructive facts of Weil descent on elliptic curves, HP Labs Tech. Report, HPL-2000-10.

[18] A. Joux and K. Nguyen, Separating decision Diffie-Hellman from Diffie-Hellman in cryptographic groups, 2001. Available at `http://eprint.iacr.org`

[19] B. Kaliski, Key Encapsulation: An Emerging Paradigm for Public-Key Cryptography, *RSA Conference 2002, Japan*, May 2002, Conference Note.

[20] N. Koblitz, Elliptic curve cryptosystems, *Math. Comp.*, **48** (1987), 203-209.

[21] U. Maurer, Towards the equivalence of breaking the Diffie-Hellman protocol and computing discrete logarithms, *Advances in Cryptology – CRYPTO '94*, LNCS **839** (1994), Springer-Verlag, 271-281.

[22] A. Menezes, T. Okamoto and S. Vanstone, Reducing elliptic curve logarithms to logarithms in a finite field, *Proceedings of the 22nd Annual ACM Symposium on the Theory of Computing*, 1991, 80-89.

[23] M. Naor and M. Yung, Universal one-way hash functions and their cryptographic applications, *Proceedings of the 21st Annual ACM Symposium on Theory of Computing*, 1989.

[24] NTT Labs, Specification of PSEC: Provably Secure Elliptic Curve Encryption Scheme, 2000. Available at `http://info.isl.ntt.co.jp/psec/CRYPTREC/index-e.html`

[25] NTT Labs, Self Evaluation of PSEC: Provably Secure Elliptic Curve Encryption Scheme, 2000. Available at `http://info.isl.ntt.co.jp/psec/CRYPTREC/index-e.html`

[26] S. C. Pohlig and M. E. Hellman, An improved algorithm for computing logarithms over GF(p) and its cryptographic significance, *IEEE Transactions on Information Theory*, IT-**24** (1978), 106-110.

[27] D. Pointcheval and T. Okamoto, The GAP problems: A new class of problems for the security of cryptographic schemes, *Proceedings of the 2001 International Workshop on Practice and Theory in Public Key Cryptography (PKC 2001)*, LNCS **1992** (2001), 104-118.

[28] J. Pollard, Monte Carlo methods for index computation (mod p), *Math. Comp.*, **32** (1978), 918-924.

[29] T. Satoh and K. Araki, Fermat quotients and the polynomial time discrete log algorithm for anomalous elliptic curves, *Commetarii Math. Univ. St. Pauli.*, Vol. 47 (1998), 81-92.

[30] I. A. Semaev, Evaluation of discrete logarithms in a group of p-torsion points of an elliptic curve in characteristic p, *Math. Comp.*, **67** (1998), 353-356.

[31] V.Shoup, Lower Bounds for Discrete Logarithms and Related Problems, *Advances in Cryptology – Eurocrypt '97*, LNCS **1233** (1997), Springer-Verlag, 256-266.

[32] V.Shoup, A Proposal for an ISO Standard for Public Key Encryption (v. 2.1), ISO/IEC JTC1/SC27, N2563, 2001. Available at `http://shoup.net/papers/` or `http://eprint.iacr.org/`

[33] N. P. Smart, The discrete logarithm problem on elliptic curves of trace one, *J. of Cryptology*, Vol. 12 (1999), No. 3, 193-196.

[34] Standard for Efficient Cryptography Group, *SEC 1: Elliptic Curve Cryptography*, ver.1.0, 2000. Available at `http://www.secg.org`

[35] Standard for Efficient Cryptography Group, *SEC 2: Recommended Elliptic Curve Domain Parameters*, 2000. Available at `http://www.secg.org`.