2004－ＣＳＥＣ－26 （9）
2004／7／20

あるタイプの公開鍵暗号変換方式とその関連問題の考察

藤崎英一郎†

† 日本電信電話株式会社 〒239-0847 神奈川県横須賀市光の丘1-1
E-mail: †fujisaki@isl.ntt.co.jp

あらまし ISOで標準化が議論されているECIES-HC、RSA-HC[5]を包含する汎用的な（選択暗号文攻撃安全な）公開鍵暗号への変換方式を考える。この種の方式は、一般に復号の計算量を少なく抑えられるという利点があるといわれている。しかし、本稿では変換方式の安全性を厳密に考えることにより、復号の手順において、従来不必要とも考えられている検査が、必要であることを示す。すなわち、これらの方式は従来考えられているより、一般的に復号の計算量は悪くなることを示す。

キーワード ECIES-KEM, RSA-KEM, OW-PCA, 復号の計算量.

A note on a decryption problem for some type of generic conversion method to secure encryption

Eiichiro FUJISAKI†

† NTT Labs, 1-1 Hikarinooka Yokosuka-shi Kanagawa 239-0847 JAPAN
E-mail: †fujisaki@isl.ntt.co.jp

Abstract We present a generic method to construct secure hybrid encryption schemes (in the random oracle model), including as special cases ECIES-HC and RSA-HC argued now in ISO/IEC 18033-2[5]. In general, a hybrid encryption scheme transformed via this kind of method can enjoy a "better decryption time". However, we show that the decryption of this type can be cost more than expected in the "folklore", when giving a rigorous security proof to it.

Key words ECIES-KEM, RSA-KEM, OW-PCA, decryption time.

1. はじめに

1.1 DH型（El Gamal型）ハイブリッド暗号の復号処理

Diffie-Hellman 鍵配送[2]を利用した公開鍵暗号方式及びハイブリッド暗号方式はこれまで多々提案されている。典型的な方式として、現在ISO 18033-2[5]で標準化が進んでいるECIES-HCがある。ECIES-HCは、鍵配送方式ECIES-KEM（KEM: Key encapsulation mechanism）と、共通鍵暗号のDEM（Data encapsulation mechanism）の組み合わせで定義されるハイブリッド暗号方式である。ECIES-KEMは次のようなアルゴリズムの組(ECIES-KEM.KGE,ECIES-KEM.ENC,ECIES-KEM.DEC)で定義される：(g,G)を、Gを素数位数の巡回群、gをGの生成元とする共通パラーメータ。$H:\{0,1\}^*\times\{0,1\}^*\to\{0,1\}^k$を適当なハッシュ関数とする（群$G$の元からビット列($\in\{0,1\}^*$)への適切なエンコードが存在する）。

- ECIES-KEM.KGEは、（セキュリティパラメータk）を入力に取り、内部で適当な乱数xと$h=g^x$を計算し、公開鍵$pk=(g,h,G)$, 秘密鍵$sk=(pk,x)$を出力するアルゴリズム。

$(pk,sk)\leftarrow$ ECIES-KEM.KGE(1^k).

- ECIES-KEM.ENCは、公開鍵pkを取り、内部で適当な乱数rを生成し、$c=g^r$と$K=H(g^r,h^r)$を出力するアルゴリズム

$(c,K)\leftarrow$ ECIES-KEM.ENC$(pk,1^k)$.

- ECIES-KEM.DECは、秘密鍵skとcを取り、$c\in G$であれば、$K=H(c,c^x)$を出力するアルゴリズム。

$K\leftarrow$ ECIES-KEM.DEC$(sk,c,1^k)$.

ECIES-KEMで生成されたセッション鍵Kを、任意の選択暗号文攻撃安全な秘密鍵暗号DEMと組み合わせたハイブリッド暗号ECIES-HCは、Gap Diffie-Hellman仮定（GDH仮定）[3]とハッシュ関数Hがランダムオラクルであるという仮定の下

で、(公開鍵暗号として) 選択暗号文攻撃安全 (IND-CCA2 [1]) であることが証明されている [4]。

さてここで、ECIES-KEM の復号アルゴリズム ECIES-KEM.DEC 内の "$c \in G$" という検証に注目する。この検証は g^r を計算するのと同等の手間を有する。もしこの検証を省略できるのであれば、復号処理はそれだけ高速になる。

例えば、g^r をハッシュ関数の入力に含めない場合 (すなわち、$K = H(h^r)$ の場合)、上記検証は、次のような反例から明らかに省略できない。

[$K = H(h^r)$ とした時のハイブリッド暗号に対する選択暗号文攻撃を考える] G を、ある体 K の乗法群 K^* の有限部分群とする (すなわち、$G \subset K^*$)。$(g^r, E_K(m))$ (セッション鍵 $K = H(h^r)$, 共通鍵暗号 $E_K(\cdot)$) をチャレンジ暗号文とする。選択暗号文攻撃として、$(-g^r, E_K(m))$ を復号オラクルに送る。復号時に「$c \in G$ の検査をしない」場合、$(-g^r)^x = (-1)^x h^r$ より、(x は鍵生成アルゴリズムによりランダムに選ばれているので) ほぼ 1/2 の確率で、$K = H((-c)^x)$ が成り立ち、攻撃者は平文 m を得ることができてしまう。

この攻撃を防ぐには、ハッシュ関数への入力値に g^r を加えて $K = H(g^r, h^r)$ とすればよい。こうすれば、(復号時の)「$c \in G$ の検証」を省略していても、この攻撃がうまくいかないことは明らかである。ではそれ以外の攻撃に対してはどうであろうか?

定説、風聞 (?) として、$K = H(g^r, h^r)$ とすれば、「$c \in G$ の検証」は不要であるという話を聞く。本稿では、その件に関して検討する。

1.2 RSA 型ハイブリッド暗号の一般化

ISO [5] で標準化が議論されている RSA-HC を考える。ECIES-HC 同様、RSA-HC は、鍵配送方式 RSA-KEM と、共通鍵暗号の DEM の組み合わせで定義されるハイブリッド暗号方式である。RSA-HC は、RSA 問題の一方向性の仮定 (RSA 仮定) とランダムオラクル仮定の下で、(公開鍵暗号として) 選択暗号文攻撃安全である [4]。

RSA 関数を任意の (落し戸付き) 一方向性置換に置き換える。当然のように、この一般化は成り立つ。では、任意の確定的な (deterministic) 公開鍵暗号に置き換えるとどうであろうか? さらに、任意の (確率的な場合も含む) 公開鍵暗号に置き換えた場合はどうであろうか?

1.3 本稿の結果

ISO [5] で標準化が議論されている ECIES-KEM、RSA-KEM を包含する汎用的な KEM への変換方式を考える。この方式を定義するにあたって、原始的鍵配送方式なるものを定義する。これは、任意の公開鍵暗号と DH 型鍵配送方式を同時に定義するためのものである。その後、原始的鍵配送方式を KEM に変換する。公開鍵暗号方式の場合と同様、この原始的鍵配送方式に対して、OW-PCA に類似のクラス (実際、そのうちのもっとも弱いクラスは OW-CPA に一致する) を定義し、これとランダムオラクル仮定のもと安全な [4]KEM であることを証明する。よって、任意の安全な共通鍵暗号と組み合わせることで、安全な公開鍵暗号 (ハイブリッド暗号) が構成できる。

この証明の過程においていくつかのことが分かる。一つは ECIES-KEM において、GapDH 仮定の下で安全性を証明するのであれば前述の復号時の $c \in G$ の検証は必要であることがわかる。

もう一つは意外なことに、RSA を任意の確定的な (deterministic) 公開鍵暗号に置き換えて RSA-KEM の代わりになるものを作った場合、一般には復号効率を著しく落さなければいけないことがわかる。

2. 準　　備

本稿では、Shoup の KEM, ハイブリッド暗号の枠組みを利用する。そのため幾つか準備をしておく。以下の定義や定理は全て [4] によっている。

a) KEM: 鍵配送方式

[定義 2.1] 鍵配送方式 KEM= (KEM.KGE, KEM.ENC, KEM.DEC) は次のような条件を満たすアルゴリズムの組である。

- KEM.KGE は、セキュリティパラメータ k を入力に取り、公開鍵 pk, 秘密鍵 sk を出力する (k に関する) 確率的多項式時間アルゴリズム。

$(pk, sk) \leftarrow \mathtt{KEM.KGE}(1^k)$.

- KEM.ENC は、公開鍵 pk を取り暗号文 $c \in \{0,1\}^k$ とセッション鍵 $K \in \{0,1\}^k$ を出力する (k に関する) 確率的多項式時間アルゴリズム。

$(c, K) \leftarrow \mathtt{KEM.ENC}(pk, 1^k)$.

- KEM.DEC は、秘密鍵 sk とビット列 $c' \in \{0,1\}^k$ を取り、ビット列 $K' \in \{0,1\}^k$ を出力する (k に関する) 確定的多項式時間アルゴリズム。

$K' \leftarrow \mathtt{KEM.DEC}(sk, c, 1^k)$.

ただし、$(pk, sk) \leftarrow \mathtt{KEM.KGE}(1^k)$, $(c, K) \leftarrow \mathtt{KEM.ENC}(pk, 1^k)$ であるならば、常に $\mathtt{KEM.DEC}(sk, c, 1^k) = K$ であることを必要とする。

b) KEM の安全性のモデル

次に安全な KEM というものを定義する。

次のような敵 A の攻撃環境を考える。

(1) セキュリティパラメータ k を固定。

(2) KEM.KGE, KEM.ENC を動作させ、$(pk, sk) \leftarrow \mathtt{KEM.KGE}(1^k)$、$(c^*, K^*) \leftarrow \mathtt{KEM.ENC}(pk, 1^k)$ をそれぞれ出力させる。

(3) $b \in \{0,1\}$ と $K' \in \{0,1\}^k$ をランダムに選び、$K_b := K^*$、$K_{\bar{b}} := K'$ と代入する。

(4) A に (pk, c^*, K_0, K_1) を入力する。

(5) A は、$\mathtt{KEM.DEC}(sk, \cdot)$ オラクルにアクセスできる。ただし、c^* を質問することだけは禁止とする。

(6) A は、ビット $b' \in \{0,1\}$ を出力して停止する。

A の KEM に対するアドバンテージを $\mathrm{Adv}_{\mathtt{KEM}} A(k) \triangleq 2\Pr[b = b'] - 1$ で定義する。確率は上記の攻撃環境と A によって決定される。

[定義 2.2] KEM が安全であるとは、任意の（k に関する）多項式時間アルゴリズム A に対して、$\mathrm{Adv}_{\mathbf{KEM}}A(k)$ が（k に関して）無視できる確率であるときを言う。

c） ハイブリッド暗号

$\Pi = (\mathcal{E}, \mathcal{D})$ を $\{0,1\}^k$ を鍵空間とする共通鍵暗号とする。KEM で作ったセッション鍵 K により平文 $m \in \{0,1\}^*$ を暗号化した暗号文を $E_K(m)$ とする。KEM と Π からなるハイブリッド暗号 $\Pi' = (\mathcal{K}', \mathcal{E}', \mathcal{D}')$ は次のように（自明に）定義される。

鍵生成アルゴリズム $\mathcal{K}'$: 入力： 1^k; $(pk, sk) \leftarrow \mathtt{KEM.KGE}(1^k)$; 出力： (pk, sk).

暗号化アルゴリズム $\mathcal{E}'$: 入力： (pk, m) $(m \in \{0,1\}^*)$; $(c, K) \leftarrow \mathtt{KEM.ENC}(pk, 1^k)$; $e \leftarrow E_K(m)$; 出力： (c, e).

復号化アルゴリズム $\mathcal{D}'$: 入力： (sk, C) $(C \in \{0,1\}^*)$; C をうまく $c||e$ に分解できない場合は $\perp$ を返して停止する; $K' \leftarrow \mathtt{KEM.DEC}(sk, c, 1^k)$; 出力：$\mathcal{D}'_{K'}(e)$.

[定理 2.3] 鍵配送方式 KEM が安全で、共通鍵暗号 Π が選択暗号文攻撃安全とする。そのときハイブリッド暗号 Π' は選択暗号文攻撃安全である

詳細は、文献[4] を参照のこと。

3. 提案方式

3.1 原始的鍵配送方式

$F : \{0,1\}^* \to \{0,1\}^*$, $G : \{0,1\}^* \to \{0,1\}^*$, $D : \{0,1\}^* \to \{0,1\}^* \cup \{\perp\}$ を入力サイズに対して多項式時間計算可能な関数。$\{X\}_{k \in \mathbf{N}}$ を、各 k に対して定義された集合 $X \subset \{0,1\}^k$ の族とする。さらに、X の要素を k に対する多項式時間で一様に抽出可能かつ、$x \in \{0,1\}^k$ が $x \in X$ であるかを k に対する多項式時間で検証可能とする。F, G の入力を X に制限した関数を、それぞれ $f(\triangleq F|_X)$, $g(\triangleq G|_X)$ と書く。f, g の値域を、$Y(\triangleq Im(f))$, $Z(\triangleq Im(g))$ とする。$Y, Z \subset \{0,1\}^k$ としても、一般性を失うことは無いのでそうしておく。D の入力を $\{0,1\}^k$ に制限した関数を、$d(\triangleq D|_{\{0,1\}^k})$ と書く。以上のような関数の組 (F, G, D) に対して、原始的鍵配送方式を次のように定義する。

[定義 3.1] 上記のような関数の組 (F, G, D) が、任意の $k \in \mathbf{N}$ に対して、$x \in X$ ならば常に $d(f(x)) = g(x)$ であるとき、「原始的鍵配送方式」であるという。

[注意 3.2] 原始的鍵配送方式 (F, G, D) は、（落し戸付）一方向性置換、公開鍵暗号、DH 鍵配送の一般化になっている。

- $X = Y = Z$、$g(x) = x$ とすると、(f, d) は置換とその逆置換。
- $X = M \times R$、$Z = M$、$g(m, r) = m$ とすると（$m \in M$, $r \in R$）、(f, d) は暗号化アルゴリズムと復号化アルゴリズムとする公開鍵暗号。
- G を、素数位数 q の部分群、$g, h \in G$ として、$X = \mathbf{Z}/q\mathbf{Z}$、$Y = Z = G$、$f(x) = g^x$、$g(x) = h^x$ とおけば、DH 鍵配送。

[定義 3.3] 任意の k-ビット列 $y \in \{0,1\}^k$ が与えられた時、$y \in Y$ を（k に関する）多項式時間で検証可ならば、原始的鍵配送方式 (F, G, D) は、特に well-formed であるということにする。

通常の公開鍵暗号は必ずしも well-formed の性質を備えていない。一方、DH 鍵配送は、well-formed である。

3.2 検証関数 CO_i

原始的鍵配送方式 (F, G, D) に対する関数 $CO_i : \{0,1\}^* \times \{0,1\}^* \to \{0,1\}$ $(i \in \{0,1,2\})$ を次のように定義する。

[定義 3.4] $\forall k \in \mathbb{N}$、$\forall c, t \in \{0,1\}^k$ に対して、

（1） $CO_0(c,t) = 1$ iff $\exists x \in X\,[c = f(x)$ and $t = g(x)]$.

（2） $CO_1(c,t) = 1$ iff $d(c) = t$ and $t \in Z$.

（3） $CO_2(c,t) = 1$ iff $d(c) = t$.

たとえば、(F, G, D) が DH 鍵配送方式の場合、CO_0 は DDH 関係を判定する関数に等しい。下の補題でわかるが、CO_0,CO_1, CO_2 となるにつれチェックが甘くなっていく。

[補題 3.5] $CO_0(c,t) = 1$ iff $d(c) = t$ and $c \in Y$.

証明略。

さらに、

[補題 3.6] $d(c) = t$ and $c \in Y$ iff $d(c) = t$, $c \in Y$ and $t \in Z$.

証明略。

(F, G, D) が well-formed でないとすると、CO_0 のチェックは、d を持っていたとしても多項式時間アルゴリズムには実行できないことに注意せよ。

3.3 OW-PCA

原始的鍵配送方式 (F, G, D) と上記検証関数 CO_i に対して、安全性のクラス $\mathtt{OW\text{-}PCA}_i$ $(i \in \{0,1,2\})$ を次のように定義する。

次のような敵 A の攻撃環境を考える。

（1） セキュリティパラメータ k を固定（(f, g, d, X, Y, Z) が決まる）。

（2） $x^* \in X$ を一様に抽出し、$c^* = f(x^*)$ を計算する。

（3） A に (f, g, X, Y, Z, c^*) を入力する。

（4） A は、検証関数 CO_i オラクルにアクセスできる。

（5） A は、ビット列 t' を出力して停止する。

A の (F, G, D) に対するアドバンテージを $\mathrm{Adv}^i_{(F,G,D)}A(k) \triangleq \Pr[t' = g(x^*)]$ で定義する。確率は上記の攻撃環境と A によって決定される。

[定義 3.7] 原始的鍵配送方式 (F, G, D) が $\mathtt{OW\text{-}PCA}_i$ であるとは、任意の（k に関する）多項式時間アルゴリズム A に対して、$\mathrm{Adv}^i_{(F,G,D)}A(k)$ が（k に関して）無視できる確率であるときを言う。

[補題 3.8]

- $\mathtt{OW\text{-}PCA}_0$, $\mathtt{OW\text{-}PCA}_1$, $\mathtt{OW\text{-}PCA}_2$ の順に強い仮定になる。
- 原始的鍵配送方式 (F, G, D) が DH 鍵配送方式であるとき、$\mathtt{OW\text{-}PCA}_0$ 仮定と Gap DH (GDH) 仮定は等しい。
- 原始的鍵配送方式 (F, G, D) が確定的 (deterministic) 公開鍵暗号だとすると、$\mathtt{OW\text{-}PCA}_0$ 仮定と、公開鍵暗号の一方向性 (OW) 仮定は等価である（$\mathtt{OW\text{-}PCA}_1$, $\mathtt{OW\text{-}PCA}_2$ は OW より強い仮定となる）。

3.4 (F, G, D)-KEM 方式

原始的鍵配送方式 (F, G, D) とハッシュ関数 $H : \{0,1\}^* \times \{0,1\}^* \to \{0,1\}^k$ から次のような KEM を作る。

- (F, G, D)-KEM.KGE は、セキュリティパラメータ k を

入力に取り、公開鍵 $pk = (f, g, X, Y, Z)$, 秘密鍵 $sk = (pk, d)$ を出力するアルゴリズム。

- (F, G, D)-KEM.ENC は、公開鍵 pk を取り、ランダムに $x \in X$ を選び、暗号文 $c = f(x)$ と鍵 $K = H(f(x), g(x))$ を出力するアルゴリズム。
- (F, G, D)-KEM.DEC は、三種類ある。

(1) 秘密鍵 sk とビット列 $c' \in \{0,1\}^k$ を取り、$c' \in Y$ ならば、$K' = H(c', d(c'))$ を出力し、それ以外は $\perp$ を返すアルゴリズム。

(2) 秘密鍵 sk とビット列 $c' \in \{0,1\}^k$ を取り、$d(c') \in Z$ なら、$K' = H(c', d(c'))$ を出力し、それ以外は $\perp$ を返すアルゴリズム。

(3) 秘密鍵 sk とビット列 $c' \in \{0,1\}^k$ を取り、$d(c') \neq \perp$ なら、$K' = H(c', d(c'))$ を出力し、それ以外は $\perp$ を返すアルゴリズム。

復号処理に (1)-(3) のどれを選んだかに応じて、上から (F, G, D)-KEM$_0$, (F, G, D)-KEM$_1$, (F, G, D)-KEM$_2$ をそれぞれ定義する。

以下が本論文の主定理である。

[定理 3.9] 各 $i \in \{0, 1, 2\}$ に対して、(F, G, D)-KEM$_i$ は、OW-PCA$_i$ 仮定とランダムオラクル仮定の下で、安全である。

詳細は発表および、extended abstract で。

4. 復号処理の計算量についての考察およびまとめ

(F, G, D) が DH 鍵配送方式のとき OW-PCA$_0$ 仮定は、Gap DH (GDH) 仮定に等しいので、定理 3.9 の条件を満たすためには、$c \in G$ の検査が必要であることがわかる。よって、ECIES-KEM の復号処理では、$c \in G$ の検査分の計算量を省くことは出来ない。

(F, G, D) が確定的公開鍵暗号とする。すると OW-PCA$_0$ 仮定と公開鍵暗号の一方向性 (OW) 仮定が等しくなる。よって、(F, G, D)-KEM を一方向性仮定の下安全にするには、復号アルゴリズムにおいて、$c' \in Y$ の検査が必要となる。これは一般の確定的公開鍵暗号の場合、$c' = f(d(c'))$ によって検査する必要があり、その分復号の効率が悪くなる。

さらに (F, G, D) が一般の確率的公開鍵暗号とする。この時、(F, G, D) が well-formed でないばあい、いわゆる OW-CPA 仮定の下では安全性の証明がつかない。それより多少強い OW-PCA$_1$ 仮定が必要になる。同様のことが文献 [3] にもいえると考えられる。一般の確率的公開鍵暗号において、OW-PCA の定義が今まできちんとされてこなかった可能性も十分あるので気をつける必要がある。

定理 3.9 はあくまで十分条件を示しているだけなので、弱い OW-PCA$_0$ 仮定のもと、より復号効率の良い (F, G, D)-KEM$_1$ の安全性を示すことが出来るかもしれない (とてもそう思えないが)。一方、OW-PCA$_1$ 仮定のもとでは、一般に (F, G, D)-KEM$_2$ の安全性は成り立たないという強い傍証が存在する。これについては発表もしくは extended abstract に詳細を記述する予定である。

文　献

[1] M. Bellare, A. Desai, D. Pointcheval, and P. Rogaway. Relations among notions of security for public-key encryption schemes. In H. Krawczyk, editor, *Advances in Cryptology — CRYPTO'98*, volume 1462 of *Lecture Notes in Computer Science*, pages 26–45. Springer-Verlag, 1998.

[2] W. Diffie and M. E. Hellman. New directions in cryptography. *IEEE Transactions on Information Theory*, IT-22(6):644–654, 1976.

[3] T. Okamoto and D. Pointcheval. REACT: Rapid enhanced-security asymmetric cryptosystem transform. In CT – RSA '2001, volume 2020 of *LNCS*, pages 159–175. SV, 2001.

[4] V. Shoup. A proposal for an ISO standard for public key encryption. Technical report, December 2001. Cryptology ePrint Archive, Report 2001/112 `http://eprint.iacr.org`.

[5] V. Shoup. ISO/IEC 18033-2: Information technology – Security techniques – Encryption algorithms – Part 2: Asymmetric Ciphers, January 2004.