I240 暗号理論 2019
公開鍵暗号と署名（2）

2019/11/22 講師 藤崎

1 はじめに

Cramer-Shoup 公開鍵暗号を紹介し、これが IND-CCA 安全を満たすことを証明する。

（追記）証明の順番を間違えたので、ハッシュ関数の仮定が TCR 安全性ではなく、CR 安全性になってしまっています。今回は CR 安全性にして、資料を修正しておきます。その他、微細な修正（2倍を忘れたとか)。

2 公開鍵暗号

2.1 公開鍵暗号の定義

$\Pi = (\mathbf{K}, \mathbf{E}, \mathbf{D})$ はセキュリティパラメータ $\kappa \in \mathbb{N}$ に依存する三つのアルゴリズムの組みであり次のように定義される：

- 鍵生成アルゴリズム $\mathbf{K}$: 1^κ を入力としてとり、(pk, sk) を出力する確率的多項式時間アルゴリズム。この試行を $(pk, sk) \leftarrow \mathbf{K}(1^\kappa)$ と書く。(pk, sk) をそれぞれ公開鍵、秘密鍵と呼ぶ。
- 暗号化アルゴリズム $\mathbf{E}$: 公開鍵 pk と $m \in \mathcal{M}$ を入力としてとり、ct を出力する確率的多項式時間アルゴリズム。この試行を $\mathsf{ct} \leftarrow \mathbf{E}_{pk}(m)$ と書く。m を平文、ct を暗号文と呼ぶ。
- 復号化アルゴリズム $\mathbf{D}$: 秘密鍵 sk と暗号文 $\mathsf{ct} \in \mathcal{C}$ を入力としてとり m を出力する確定的多項式時間アルゴリズム。この試行を $m \leftarrow \mathbf{D}_{sk}(\mathsf{ct})$ と書く。

平文空間 $\mathcal{M}$ と暗号文空間 $\mathcal{C}$ は pk に依存して $\{0,1\}^*$ の部分集合として一意に定まるものとする。

定義 1 $\Pi = (\mathbf{K}, \mathbf{E}, \mathbf{D})$ が、十分大きな全てのセキュリティパラメータ $\kappa \in \mathbb{N}$ に対して、$(pk, sk) \in \mathbf{K}(1^\kappa)$, $m \in \mathcal{M}$, $\mathsf{ct} \in \mathbf{E}_{pk}(m)$ なら $\mathbf{D}_{sk}(\mathsf{ct}) = m$ を常に満足する **(Correctness)** とき、Π を**公開鍵暗号**とよぶ。

2.2 公開鍵暗号の安全性のクラス

公開鍵暗号の安全性は, 攻撃者 (adversary) の解読目標（「部分解読」か「完全解読」等）への耐性と、攻撃者の利用可能な環境（選択平文攻撃、選択暗号文攻撃）の2つの独立な要素の組み合わせによって定義される。「解読目標耐性」には、「一方向性（ one-wayness (OW), 完全解読不可)」「識別不可能性（ indistinguishability (IND), いかなる部分解読も不可)」「頑強性（ non-malleability (NM))」などが存在する。一方、攻撃者の環境には「受動的攻撃（選択平文攻撃 (chosen-plaintext attack (CPA)) 」「非適応的選択暗号文攻撃 (non-adaptive chosen-ciphertext attack or lunch-time attack (CCA1)) 」「適応的選択暗号文攻撃 (adaptive chosen-ciphertext attack (CCA2)) 」などがある。

以下では、IND-CPA 安全性（選択平文攻撃に対する識別不可能性）と IND-CCA2 安全性（適応的選択暗号文攻撃に対する識別不可能性）について説明する。**IND-CCA1 安全性を扱わないので、IND-CCA2 安全性を紛れがないときは単に IND-CCA 安全性とよぶことにする。**

■識別不可能性 (indistinguishability)　公開鍵暗号が暗号が通信情報を秘匿するものであることを鑑みると、一方向性は暗号の安全性を定義するのに十分な概念とは言えない。公開鍵暗号のいかなる部分情報解読も不可であることを示すのに次のようなゲームが利用される。

[公開鍵暗号 Π の識別ゲーム]　ゲームは, 攻撃者 A と攻撃者に問題を与える挑戦者 S からなる。

1. 挑戦者 S は, $\mathbf{K}$ を起動して $(pk, sk) \leftarrow \mathbf{K}(1^\kappa)$ を得た後, 攻撃者 A に pk を与える。
2. 攻撃者 A は, 二つの平文 $m_0, m_1 \in \mathcal{M}$ を出力する。
3. 挑戦者 S は, コイン $b^* \in \{0,1\}$ をふり, m_0, m_1 のどちらを暗号化するか決める。
4. 攻撃者 A に $\mathsf{ct}^* = \mathbf{E}_{pk}(m_{b^*})$ を入力する。
5. 攻撃者 A は, ビット $b \in \{0,1\}$ を出力する。
6. $b = b^*$ の時, A の勝利、それ以外のときは S の勝利とする。

仮に, Π がいかなる部分情報も漏らさないのであれば, A が勝利できる確率は、高々 $1/2$ である。A の $1/2$ を超えて勝利できる確率を（正規化したものを）A の成功確率（アドバンテージ）として定義する。しかしまず, このゲームにバリエーションを与えてやることを考える。

■選択平文攻撃・(適応的) 選択暗号文攻撃　A が, 暗号化オラクル $\mathbf{E}_{pk}$、復号化オラクル $\mathbf{D}_{sk}$ を利用できるかで、上記のゲームは二つのバリエーションができる。暗号化オラクル $\mathbf{E}_{pk}$ を利用できる環境を, 選択平文攻撃という。一方, 復号化オラクル $\mathbf{D}_{sk}$ を利用できる環境を, 選択暗号文攻撃という。

公開鍵暗号では, A は pk を知っているので, 自分で $\mathbf{E}_{pk}$ を利用して任意の平文 m を暗号化できる。よって選択平文攻撃はもっとも弱い A の攻撃法である。

選択暗号文攻撃では, A は上記のゲームにおいていかなる時でも、復号化オラクル $\mathbf{D}_{sk}$ を利用できる。唯一の制限は, ステップ (4) で入力されたチャレンジ暗号文 $\mathbf{E}_{pk}(m_{b^*})$ を復号化オラクルに質問してはいけないということである。このようなチャレンジ暗号文をもらった後、復号オラクルにアクセスできる攻撃法を適応的選択暗号文攻撃 (adaptive chosen-ciphertext attack (CCA2)) と呼ぶが、ここでは単に選択暗号文攻撃 (chosen-ciphertext attack (CCA)) と呼ぶことにする。

■IND-CPA-安全, IND-CCA-安全　以下に、INDATK ゲームで攻撃者 $A = (A_1, A_2)$ が正しく b^* を推測できた場合、1 を出力するような試行（確率変数）を定義する。

$\mathsf{Expt}^{\mathsf{ind\text{-}atk}}_{\Pi,A}(\kappa)$:
$(pk, sk) \leftarrow \mathbf{K}(1^\kappa)$
$(m_0, m_1, st) \leftarrow A_1^{O_1}(pk)$
$b^* \leftarrow \{0,1\}; \mathsf{ct}^* \leftarrow \mathbf{E}(pk, m_{b^*})$
$b \leftarrow A_2^{O_2}(st, \mathsf{ct}^*)$
If $b = b^*$,　return 1 else return 0.

図 1　Security Experiment for IND-ATK

と定義する。上記確率は $\mathbf{K}, \mathbf{E}_{pk}, A$（の内部コインに）依存する。

- ATK=CPA (IND-CPA Game): $O_1 = O_2 = \perp$. A はオラクルに質問できない。
- ATK=CCA1 (IND-CCA1 Game): $O_1 = \mathbf{D}_{sk}(\cdot)$, $O_2 = \perp$. A はチャレンジ暗号文をもらった後はオラクルに質問できない。

- ATK=CCA2 (IND-CCA2 Game): $O_1 = O_2 = \mathbf{D}_{sk}(\cdot)$, A はチャレンジ暗号文をもらった後も**チャレンジ暗号文以外であれば**オラクルに質問できる。

攻撃者 A の Π に対する IND-ATK 攻撃成功確率（アドバンテージ）を

$$\mathsf{Adv}^{\mathsf{ind\text{-}atk}}_{\Pi,A}(\kappa) := \left| 2\Pr[\mathsf{Expt}^{\mathsf{ind\text{-}atk}}_{\Pi,A}(\kappa) = 1] - 1 \right|$$

と定義する。

今回、IND-CCA1 はあまり考えないので、**IND-CCA2 を IND-CCA と書き**、IND-CPA 安全性と IND-CCA 安全性についてのみ定義する。

定義 2（IND-CPA, IND-CCA 安全性） 復号オラクル（O_1 オラクル、O_2 オラクル）に最大 q 回質問する任意の t-時間アルゴリズム（族）の攻撃者 A を考える。この A に対して、十分大きな全ての $\kappa \in \mathbb{N}$ で, $\mathsf{Adv}^{\mathsf{ind\text{-}cca}}_{A,\Pi}(\kappa) \leq \epsilon$ が成立するとき, 公開鍵暗号 Π は, $(t(\kappa), q(\kappa), \epsilon(\kappa))$-IND-CCA-安全であるとよぶ。$q(\kappa) = 0$ のとき、特に (t, ϵ)-IND-CPA-安全とよぶ。特に、$t, q = O(\mathrm{poly}(\kappa))$, $\epsilon(\kappa) = \mathsf{negl}(\kappa)$ ならば、Π は IND-CCA-安全（$q(\kappa) = 0$ なら IND-CPA-安全）であるという。

次のような試行を考える。

$\mathsf{Expt}^{\mathsf{ind\text{-}atk}\text{-}b^*}_{\Pi,A}(\kappa)$
$(pk, sk) \leftarrow \mathbf{K}(1^\kappa)$; $\quad (m_0, m_1, st) \leftarrow A_1^{O_1}(pk)$
$\mathsf{ct}^* \leftarrow \mathbf{E}(pk, m_{b^*})$; $\quad b \leftarrow A_2^{O_2}(st, \mathsf{ct}^*)$
return bit b.

図 2 Security Experiment for IND-ATK-b^*

図 2 のように試行が定義された時、

$$\mathsf{Adv}^{\mathsf{ind\text{-}atk}}_{\Pi,A}(\kappa) = \left| \Pr[\mathsf{Expt}^{\mathsf{ind\text{-}atk}\text{-}1}_{\Pi,A}(\kappa) = 1] - \Pr[\mathsf{Expt}^{\mathsf{ind\text{-}atk}\text{-}0}_{\Pi,A}(\kappa) = 1] \right| \tag{1}$$

が成り立つ。

問題 3 式 (1) を証明せよ。

3 ハッシュ関数（族）

ハッシュ関数族 $\mathcal{H} = \{H_i\}_{i \in \mathcal{I}}$ は、（ハッシュ）関数 $H_i : \{0,1\}^* \to \{0,1\}^{p(\kappa)}$ $(i \in \mathcal{I}_\kappa)$ の族である。

ハッシュ関数族の安全性として、「ターゲット衝突困難性 (Target Collision Resistance)」、「衝突困難性 (Collision Resistance)」を定義する。

定義 4（ターゲット衝突困難性 (TCR)） $\mathcal{H} = \{H_i\}_{i \in \mathcal{I}}$ のターゲット衝突困難性を攻撃する敵 $A = (A_1, A_2)$ による攻撃成功確率を、

$$\mathsf{Adv}^{\mathsf{tcr}}_{\mathcal{H},A}(\kappa) := \Pr\left[\begin{array}{ll} (x, st) \leftarrow A_1(1^\kappa); & \\ i \leftarrow \mathcal{I}_\kappa; & \\ y \leftarrow A_2(H_i, st) & : \quad x \neq y \text{ and } H_i(x) = H_i(y) \end{array} \right]$$

と定義する。任意の t-時間の敵 A に対しても、$\mathsf{Adv}^{\mathsf{tcr}}_{\mathcal{H},A}(\kappa) \leq \epsilon$ を満足するのであれば、$\mathcal{H}$ は、(t, ϵ)-TCR であると言う。$t(\kappa) = O(\mathrm{poly}(\kappa))$, $\epsilon(\kappa) = \mathsf{negl}(\kappa)$ ならば、$\mathcal{H}$ は、ターゲット衝突困難 (TCR) であると言う。

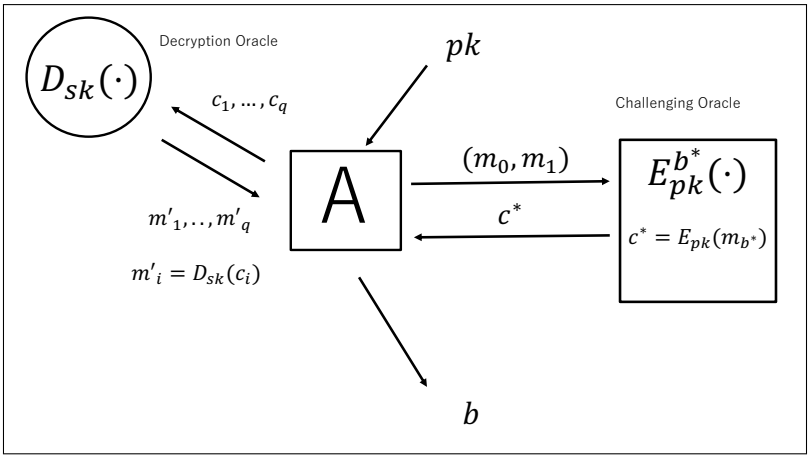

図 3　IND-CCA 安全性モデル

定義 5（衝突困難性 (CR)）　$\mathcal{H} = \{H_i\}_{i\in\mathcal{I}}$ の衝突困難性を攻撃する敵 A による攻撃成功確率を、

$$\mathsf{Adv}^{\mathsf{cr}}_{\mathcal{H},A}(\kappa) := \Pr[i \leftarrow \mathcal{I}_\kappa;\ (x,y) \leftarrow A(H_i) : x \neq y \text{ and } H_i(x) = H_i(y)]$$

と定義する。任意の t-時間の敵 A に対しても、$\mathsf{Adv}^{\mathsf{cr}}_{\mathcal{H},A}(\kappa) \leq \epsilon$ を満足するのであれば、$\mathcal{H}$ は、(t,ϵ)-CR であると言う。$t(\kappa) = O(\mathrm{poly}(\kappa))$, $\epsilon(\kappa) = \mathsf{negl}(\kappa)$ ならば、$\mathcal{H}$ は、衝突困難 (CR) であると言う。

仮定の強弱でいうと、「一方向性」＝「ターゲット衝突困難性」＜「衝突困難性」であり、ターゲット衝突困難ハッシュ関数は一方向性関数があれば構成できることが知られているが、衝突困難ハッシュ関数が一方向性関数から構成できるかは知られていない。しかし、現実に使われる SHA2,3 等のハッシュ関数などは「衝突困難性」を当然満たすべき条件としていて、仮に衝突が見つかった場合、危殆化したと判断されるはずである。

4　Cramer-Shoup 暗号

Cramer と Shoup によって提案された Cramer-Shoup 暗号（以下、CS 暗号）を紹介する [CS98, CS02, CS03]。

■HPS-CS 暗号:　$\mathsf{KGen}_{\mathsf{group}}$ を付録 付録 B で定義される DDH 仮定を満足するアルゴリズムとする。HPS-CS 暗号 $\mathsf{HPS\text{-}CS} = (\mathbf{K}, \mathbf{E}, \mathbf{D})$ は以下のようなアルゴリズムの組である。

- 鍵生成アルゴリズム $\mathbf{K}$:　入力 1^κ. 出力 (pk, sk).
  1. $H \leftarrow \mathcal{H}$;
  2. $(G_q, q) \leftarrow \mathsf{KGen}_{\mathsf{group}}(1^\kappa)$;
  3. $g \leftarrow G_q \backslash \{1\}$;
  4. $\alpha_1, \alpha_2 \leftarrow \mathbb{Z}/q\mathbb{Z}$; $g_1 := g^{\alpha_1}$; $g_2 := g^{\alpha_2}$;
  5. $z_1, z_2 \leftarrow \mathbb{Z}/q\mathbb{Z}$; $h := g_1^{z_1} g_2^{z_2}$;
  6. $x_1, x_2, y_1, y_2 \leftarrow \mathbb{Z}/q\mathbb{Z}$; $c := g_1^{x_1} g_2^{x_2}$; $d := g_1^{y_1} g_2^{y_2}$;
  7. $pk := (G_q, q, g_1, g_2, h, c, d)$; $sk := (pk, z_1, z_2, x_1, x_2, y_1, y_2)$.
- 暗号化アルゴリズム $\mathbf{E}$:　入力 (pk, m). 出力 ct.
  1. $m \notin G_q$ なら停止。それ以外は次のステップへ。
  2. $r \leftarrow \mathbb{Z}/q\mathbb{Z}$; $u_1 := g_1^r$; $u_2 := g_2^r$;

3. $e := m \cdot h^r$;
4. $t := H(u_1, u_2, e)$;
5. $v := (cd^t)^r$;
6. $\mathsf{ct} := (u_1, u_2, e, v)$.

- 復号アルゴリズム $\mathbf{D}$: 入力 (sk, ct). 出力 m.
  1. $\mathsf{ct} = (u_1, u_2, e, v) \in G_q^4$ と正しく分解できなければ、$m := \perp$ を出力し停止。それ以外は次のステップへ。
  2. $t' := H(u_1, u_2, e)$;
  3. $v \neq u_1^{x_1 + t' y_1} u_2^{x_2 + t' y_2}$ の場合、$m := \perp$ を出力し停止。それ以外は次のステップへ。
  4. $m := e \cdot (u_1^{z_1} u_2^{z_2})^{-1}$ を出力。

定理 6 $\mathcal{H}$ が $(t_{\mathsf{cr}}, \epsilon_{\mathsf{cr}})$-CR, $\mathsf{KGen}_{\mathsf{group}}$ が $(t_{\mathrm{DDH}}, \epsilon_{\mathrm{DDH}})$-DDH を満足するものとする。そのとき、公開鍵暗号 HPS-CS は、(t, Q, ϵ)-IND-CCA-安全であり、

$$t \leq \max\{t_{\mathsf{tcr}}, t_{\mathrm{DDH}} - Q \cdot T_{\mathbf{E}}(\kappa)\}$$
$$\epsilon \leq 2\epsilon_{\mathsf{tcr}} + 2\epsilon_{\mathrm{DDH}} + 2\frac{Q}{q}.$$

ここで、$T_{\mathbf{E}}(\kappa)$ は、$|q| = \kappa$ のとき暗号化アルゴリズム $\mathbf{E}_{pk}$ が暗号文を生成する実行時間とする。また、$|q| = \kappa$ より、$q \approx 2^\kappa$。

$\mathcal{H}$ が CR 安全、$\mathsf{KGen}_{\mathsf{group}}$ が DDH 安全とする。すると、$t_{\mathsf{cr}}, t_{\mathrm{DDH}} = O(\mathrm{poly}(\kappa))$ のとき、$\epsilon_{\mathsf{cr}}, \epsilon_{\mathrm{DDH}} = \mathsf{negl}(\kappa)$ が成立する。多項式時間のアルゴリズムはオラクルアクセスも多項式回しかできないので、$Q = O(\mathrm{poly}(\kappa))$ として良い。よって、$t = O(\mathrm{poly}(\kappa))$ なら、$\epsilon = \mathsf{negl}(\kappa)$ である。よって、CR 安全ハッシュ関数族と DDH 仮定を満たす群生成アルゴリズムがあるなら、HPS-CS は IND-CCA 安全。

(証明) ゲーム変換によって証明する。ゲーム G_i での敵 A が b^* を正しく推測できる確率を $p_i := \Pr_{G_i}[b = b^*]$ とする。

■Game G_0: これは、オリジナルの IND-CCA ゲームである。特にチャレンジ暗号文を作成する暗号化オラクル、復号オラクルの手順を改めて記載する。

- 暗号化オラクル $\mathbf{E}_{pk}$: 入力 (m_0, m_1). 出力 ct^*:
  1. $b^* \leftarrow \{0, 1\}$;
  2. $m_{b^*} \notin G_q$ なら停止。それ以外は次のステップへ。
  3. $r^* \leftarrow \mathbb{Z}/q\mathbb{Z}$; $u_1^* := g_1^{r^*}$; $u_2^* := g_2^{r^*}$;
  4. $\Lambda^* := h^{r^*}$;
  5. $e^* := m_{b^*} \cdot \Lambda^*$;
  6. $t^* := H(u_1^*, u_2^*, e^*)$;
  7. $v^* := (cd^{t^*})^{r^*}$;
  8. $\mathsf{ct}^* := (u_1^*, u_2^*, e^*, v^*)$.
- 復号オラクル $\mathbf{D}_{sk}$: 入力 ct. 出力 m:
  1. $\mathsf{ct} = \mathsf{ct}^*$ なら、$m := \perp$ を出力し停止。
  2. $\mathsf{ct} = (u_1, u_2, e, v) \in G_q^4$ と正しく分解できなければ、$m := \perp$ を出力し停止。それ以外は次のステップへ。

3. $t' := H(u_1, u_2, e)$;
4. $v \neq u_1^{x_1+t'y_1} u_2^{x_2+t'y_2}$ の場合、$m := \perp$ を出力し停止。それ以外は次のステップへ。
5. $m := e \cdot (u_1^{z_1} u_2^{z_2})^{-1}$ を出力。

■Game G_1: Game G_0 の暗号化オラクルの Step 4 の手順を次のように変更する。他は Game G_0 と同じとする。

- 暗号化オラクル $\mathbf{E}_{pk}$: 入力 (m_0, m_1). 出力 ct^*:
  1. $b^* \leftarrow \{0, 1\}$;
  2. $m_{b^*} \notin G_q$ なら停止。それ以外は次のステップへ。
  3. $r^* \leftarrow \mathbb{Z}/q\mathbb{Z}$; $u_1^* := g_1^{r^*}$; $u_2^* := g_2^{r^*}$;
  4. $\Lambda^* := (u_1^*)^{z_1} (u_2^*)^{z_2}$;
  5. $e^* := m_{b^*} \cdot \Lambda^*$;
  6. $t^* := H(u_1^*, u_2^*, e^*)$;
  7. $v^* := (cd^r)^{t^*}$;
  8. $\mathsf{ct}^* := (u_1^*, u_2^*, e^*, v^*)$.

ここで、$u_1^* := g_1^r$, $u_2^* := g_2^r$ であるので、$h^r = (g_1^{z_1} g_2^{z_2})^r = u_1^{z_1} u_2^{z_2}$。よって、確率変数 ct^* は、Game G_0 と Game G_1 で（分布が）同一である。よって、敵 A に与えられる情報の分布は G_0, G_1 で同一である。よって、

$$p_0 = p_1.$$

■Game G_2: Game G_1 の復号オラクルの Step 3 の手順を次のように変更する。他は Game G_1 と同じとする。

- 復号オラクル $\mathbf{D}_{sk}$: 入力 ct. 出力 m:
  1. $\mathsf{ct} = \mathsf{ct}^*$ なら、$m := \perp$ を出力し停止。
  2. $\mathsf{ct} = (u_1, u_2, e, v) \in G_q^4$ と正しく分解できなければ、$m := \perp$ を出力し停止。それ以外は次のステップへ。
  3. $t' := H(u_1, u_2, e)$. もし $t' = t^*$ であれば、（復号ではなく）ゲームを中止する。
  4. $v \neq u_1^{x_1+t'y_1} u_2^{x_2+t'y_2}$ の場合、$m := \perp$ を出力し停止。それ以外は次のステップへ。
  5. $m := e \cdot (u_1^{z_1} u_2^{z_2})^{-1}$ を出力。

Game G_2 の実行時間 $t \leq t_{\mathsf{cr}}$ とする。すると、$t' = t^*$ が起きる確率は ϵ_{cr} で抑えられる。補題 8 を使うことにより、

$$p_1 - p_2 \leq \epsilon_{\mathsf{cr}}$$

が成り立つ。

■Game G_3: Game G_2 の暗号化オラクルの Step 3 の手順を次のように変更する。他は Game G_2 と同じとする。

- 暗号化オラクル $\mathbf{E}_{pk}$: 入力 (m_0, m_1). 出力 ct^*:
  1. $b^* \leftarrow \{0, 1\}$;

2. $m_{b^*} \notin G_q$ なら停止。それ以外は次のステップへ。
3. $r_1^*, r_2^* \leftarrow \mathbb{Z}/q\mathbb{Z}$; $u_1^* := g_1^{r_1^*}$; $u_2^* := g_2^{r_2^*}$;
4. $\Lambda^* := (u_1^*)^{z_1}(u_2^*)^{z_2}$;
5. $e^* := m_{b^*} \cdot \Lambda^*$;
6. $t^* := H(u_1^*, u_2^*, e^*)$;
7. $v^* := (cd^r)^{t^*}$;
8. $\mathsf{ct}^* := (u_1^*, u_2^*, e^*, v^*)$.

G_2 では、$(g_1, g_2, u_1^*, u_2^*) \leftarrow \mathrm{DDH}(G_q)$ であり、G_3 では、$(g_1, g_2, u_1^*, u_2^*) \leftarrow G_q^4$ であることに注意せよ。G_2 と G_3 での A の成功確率の差 $p_2 - p_3$ は、DDH 問題の識別確率で抑えられる。実際、A を使った DDH 識別アルゴリズムを次のように作ることができる。

DDH 識別アルゴリズム D^A: 入力 (g_1, g_2, u_1^*, u_2^*). 出力 b.

基本 G_2 の挑戦者の役割と同じことをするが、鍵生成アルゴリズムと暗号化オラクルを以下のように変更する。暗号化オラクルを次のようにする。

- 鍵生成アルゴリズム $\mathbf{K}$: 入力 (g_1, g_2). 出力 (pk, sk).
  1. $H \leftarrow \mathcal{H}$;
  2. $z_1, z_2 \leftarrow \mathbb{Z}/q\mathbb{Z}$; $h := g_1^{z_1} g_2^{z_2}$;
  3. $x_1, x_2, y_1, y_2 \leftarrow \mathbb{Z}/q\mathbb{Z}$; $c := g_1^{x_1} g_2^{x_2}$; $d := g_1^{y_1} g_2^{y_2}$;
  4. $pk := (G_q, q, g_1, g_2, h, c, d)$; $sk := (pk, z_1, z_2, x_1, x_2, y_1, y_2)$.
- 暗号化オラクル $\mathbf{E}_{pk}$: 入力: (u_1^*, u_2^*), (m_0, m_1). 出力: ct^*.
  1. $b^* \leftarrow \{0, 1\}$;
  2. $m_{b^*} \notin G_q$ なら停止。それ以外は次のステップへ。
  3. $\Lambda^* := (u_1^*)^{z_1}(u_2^*)^{z_2}$;
  4. $e^* := m_{b^*} \cdot \Lambda^*$;
  5. $t^* := H(u_1^*, u_2^*, e^*)$;
  6. $v^* := (cd^r)^{t^*}$;
  7. $\mathsf{ct}^* := (u_1^*, u_2^*, e^*, v^*)$.

最後に、A が出力したビット b と b^* を比べて $b = b^*$ であれば、D は 1 を出力し、そうでなければ 0 を出力する。

もし、$(g_1, g_2, u_1, u_2) \in \mathrm{DDH}(G_q)$ なら、D は、Game G_2 の挑戦者と全く同じことをしており、$(g_1, g_2, u_1, u_2) \in G_q^4$ なら Game G_3 の挑戦者と全く同じことをしている。よって、D の実行時間が t_{DDH} 以下ならば、$p_2 - p_3 \leq \epsilon_{\mathrm{DDH}}$ となる。D の実行時間は $t + Q \cdot T_{\mathbf{E}}(\kappa)$ であるから、$t + Q \cdot T_{\mathbf{E}}(\kappa) \leq t_{\mathrm{DDH}}$ であれば、

$$p_2 - p_3 \leq \epsilon_{\mathrm{DDH}}.$$

■Game G_4: Game G_3 の復号オラクルに次のように一つ手順を追加する。他は Game G_3 と同じとする。

- 復号オラクル $\mathbf{D}_{sk}$: 入力 ct. 出力 m:
  1. $\mathsf{ct} = \mathsf{ct}^*$ なら、$m := \perp$ を出力し停止。
  2. $\mathsf{ct} = (u_1, u_2, e, v) \in G_q^4$ と正しく分解できなければ、$m := \perp$ を出力し停止。それ以外は次のステップへ。

3. $t' := H(u_1, u_2, e)$. もし $t' = t^*$ であれば、（復号ではなく）ゲームを中止する。
4. $v \neq u_1^{x_1+t'y_1} u_2^{x_2+t'y_2}$ の場合、$m := \perp$ を出力し停止。それ以外は次のステップへ。
5. $\log_{g_1}(u_1) \neq \log_{g_2}(u_2)$ の場合、（復号ではなく）ゲームを中止する。
6. $m := e \cdot (u_1^{z_1} u_2^{z_2})^{-1}$ を出力。

Step 5 は、挑戦者が多項式時間では実行できないが仮想的試みとして出来たとしよう。実際、G_3 と G_4 での事象の確率の違いは純粋に情報量のみの違いになるので、G_4 の挑戦者は無限の実行時間を使うと仮定してもかまわない。

今、Game G_4 において、秘密情報 (z_1, z_2) が情報理論的にどの程度敵 A に漏れているか考えてみる。g を G_q の任意の生成元とすると、$\log_g(h)$, $\alpha := \log_g(g_1)$, $\alpha_2 := \log_g(g_2)$ は公開鍵から一意に決まってしまう。よって、pk が決まると (z_1, z_2) は、A の視点から見ると

$$\log_g(h) = \alpha_1 \cdot z_1 + \alpha_2 \cdot z_2 \tag{2}$$

という直線の上を動く不定の値となる。A が復号オラクルへ $\mathsf{ct} = (u_1, u_2, e, v)$ を送り、平文 m（$m \neq \perp$）を得られたとする。$\Lambda := e/m (= u_1^{z_1} u_2^{z_2})$ と定義する。$r_1 := \log_{g_1}(u_1)$, $r_2 := \log_{g_2}(u_2)$ は、暗号文から一意に決まっているから、一回復号オラクルにアクセスすると A は新たに

$$\log_g(\Lambda) = \alpha_1 r_1 \cdot z_1 + \alpha_2 r_2 \cdot z_2 \tag{3}$$

という直線の情報を得る。ところが、Game G_4 では、$r_1 = r_2$ の場合しか直線 (3) の情報は得られない。その場合、二つは直線は同一になり、復号オラクルからの情報で、(z_1, z_2) の情報は A にとって一切増えない（(z_1, z_2) の情報量が減らない）。

一方、チャレンジ暗号文 ct^* の中の Λ^* は、（$\alpha_1, \alpha_2, r_1^*, r_2^*$ が既知の）

$$\log_g(\Lambda^*) = \alpha_1 r_1^* \cdot z_1 + \alpha_2 r_2^* \cdot z_2 \tag{4}$$

という直線により決まる。(z_1, z_2) はこの直線と直線 (2) の交点である。(z_1, z_2) は元々 $(\mathbb{Z}/q\mathbb{Z})^2$ から一様ランダムに選ばれているので、直線 (2) が与えられたときは、直線 (2) 上を一様に分布する。直線 (4) と直線 (2) は交点 (z_1, z_2) の一点で交わっており、(z_1, z_2) が直線 (2) 上を一様に分布するのであれば、$\log_g(\Lambda^*)$ も G_q 上を一様に分布する。

よって、Game G_4 では、$e^* = m_{b^*} \cdot \Lambda^*$ は m_{b^*} の one-time pad である。すなわち、m_{b^*} と e^* の分布は完全に独立になってしまうので、$b = b^*$ となる確率は 1/2 になる。よって、

$$p_4 = \frac{1}{2}$$

である後は $p_3 - p_4$ を考えれば良い。

G_4 で、復号オラクルの Step 5 で、$r_1 \neq r_2$ が起きる事象を事象 F とする。差分確率補題（補題 8）から、

$$p_3 - p_4 \leq \Pr[F]$$

である。次の補題が成り立つ。

補題 7 $\Pr[F] \leq \frac{Q}{q}$

この補題はあとで証明する。これが成り立ったとすると、$p_3 - p_4 \leq \frac{Q}{q}$. まとめると、$t \leq \max\{t_{\mathsf{tcr}}, t_{\mathrm{DDH}} - Q \cdot T_{\mathbf{E}}(\kappa)$ のとき、三角不等式により

$$\mathsf{Adv}^{\mathsf{ind\text{-}cca}}_{\mathsf{HPS\text{-}CS},A}(\kappa) = |2(p_0 - p_4)| \leq 2 \left| \sum_{i=0}^{3} |p_i - p_{i+1}| \right| = 2\epsilon_{\mathsf{cr}} + 2\epsilon_{\mathrm{DDH}} + \frac{2Q}{q}.$$

以上。 ■

(証明) [補題 7 の証明] 復号オラクルで、A から来た暗号文 (u_1, u_2, e, v) が、$v = u_1^{x_1+t'y_1} u_2^{x_2+t'y_2}$ でありながら、$r_1 \neq r_2$ のとき、事象 F が起きる。次の式を考える。

$$\begin{pmatrix} \log_g(c) \\ \log_g(d) \\ \log_g(v^*) \\ \log_g(v) \end{pmatrix} = \begin{pmatrix} \alpha_1 & \alpha_2 & 0 & 0 \\ 0 & 0 & \alpha_1 & \alpha_2 \\ \alpha_1 r_1^* & \alpha_2 r_2^* & \alpha_1 r_1^* t^* & \alpha_2 r_2^* t^* \\ \alpha_1 r_1 & \alpha_2 r_2 & \alpha_1 r_1 t' & \alpha_2 r_2 t' \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ y_1 \\ y_2 \end{pmatrix} \tag{5}$$

$(\log_g(c), \log_g(d), \log_g(v^*))$ と行列の要素は、全て公開鍵とチャレンジ暗号文から一意に決まる値であるので情報量はゼロ。よって A に取って既知の値として良い。$\mathsf{ct} \neq \mathsf{ct}^*$ かつ $t' \neq t^*$ より、式 (5) の行列は正則である。

最初の 3 つの連立一次方程式は 4 次元空間の中の直線を与え、(x_1, x_2, y_1, y_2) はその直線上一様に分布している。式 (5) の行列が正則なので、4 つ目の方程式が定める超平面とその直線は一点で交わり、$\log_g(v)$ と交点 (x_1, x_2, y_1, y_2) は一対一の対応をしているため、式 (5) を満たす $\log_g(v)$ は $\mathbb{Z}/q\mathbb{Z}$ 上を一様に分布している。よって、A が v を正しく推測できる確率は高々 $\frac{1}{q}$ である。A は復号オラクルに Q 回までアクセスできるので、ゲーム中に式 (5) を満たす v を復号オラクルに送ることができる確率は $\frac{Q}{q}$ となる。よって、

$$\Pr[F] \leq \frac{Q}{q}.$$

以上で補題が証明できた。 ■

参考文献

[CS98] Ronald Cramer and Victor Shoup. A practical public key cryptosystem provably secure against adaptive chosen ciphertext attack. In Hugo Krawczyk, editor, *CRYPTO'98*, volume 1462 of *LNCS*, pages 13–25. Springer, Heidelberg, August 1998.

[CS02] Ronald Cramer and Victor Shoup. Universal hash proofs and a paradigm for adaptive chosen ciphertext secure public-key encryption. In Lars R. Knudsen, editor, *EUROCRYPT 2002*, volume 2332 of *LNCS*, pages 45–64. Springer, Heidelberg, April / May 2002.

[CS03] Ronald Cramer and Victor Shoup. Design and analysis of practical public-key encryption schemes secure against adaptive chosen ciphertext attack. *SIAM Journal on Computing*, 33(1):167–226, 2003.

付録 A　差分確率補題（再掲載）

補題 8　A, B, F をある確率空間 (probability space) の事象 (event) とする。このとき、$A \cap \overline{F} = B \cap \overline{F}$ ならば、

$$|Pr[A] - \Pr[B]| \leq \Pr[F]$$

付録 B　離散対数問題、DH 問題、DDH 問題

B.0.1　巡回群

G を有限群とする（演算は乗法で表す）このとき、$\langle g\rangle$（$g \in G$）は、G の有限可換部分群であり、準同型定理により、$\langle g\rangle$ の位数と g の位数は一致し（それを q とおく）、

$$f : x \in \mathbb{Z}/q\mathbb{Z} \to g^x \in \langle g\rangle$$

は同型写像である（$\mathbb{Z}/q\mathbb{Z}$ は加法で定義された群）。今後、$G_q := \langle g\rangle$ と書くことにする。我々は、q が素数となるようなものに特に興味がある。Lagrange の定理により、群 G_q の部分群を H とすると、$\#H|q$. q は素数であるから、H は、$\{1\}$ または G_q のどちらかである。よって、任意の $g' \in G_q\backslash\{1\}$ は G_q の生成元であり、$\langle g'\rangle = G_q$.

具体例 1: 有限体 $\mathbb{F}_p(= \mathbb{Z}/p\mathbb{Z})$（$p$ は素数）を考える。有限体の乗法群は巡回群であるので、乗法群 $\mathbb{F}_p^\times$ はある生成元 $\tilde{g} \in \mathbb{F}_p^\times$ が存在し、$\langle\tilde{g}\rangle = \mathbb{F}_p^\times$ となる。体では、$\mathbb{F}_p^\times = \mathbb{F}_p\backslash\{0\}$ であるから、$\#\mathbb{F}_p^\times = p-1$ である。q を $q|p-1$ なる奇素数とする。すると、$g := \tilde{g}^k$（$p-1 = qk$）と置くと、$f(x) = g^x \bmod p$ は、

$$f : x \in \mathbb{Z}/q\mathbb{Z} \to g^x \in G_q(= \langle g\rangle)$$

の同型写像である。

具体例２: $E(\mathbb{F}_p)$ を、有限体 $\mathbb{F}_p$ 上の楕円曲線 E の $\mathbb{F}_p$-有理点から作られる群とする。Hasse の定理により、$E(\mathbb{F}_p)$ の位数は $p+1-2\sqrt{p} \le \#E(\mathbb{F}_p) \le p+1+2\sqrt{p}$ である。$q|\#E(\mathbb{F}_p)$ なる素数を位数とする $P \in E(\mathbb{F}_p)$ の作る巡回群を $\langle P\rangle = G_q$ と置くと、

$$f : x \in \mathbb{Z}/q\mathbb{Z} \ \to \ xP \in G_q \subset E(\mathbb{F}_p)$$

は $\mathbb{Z}/q\mathbb{Z}$ から G_q への同型写像である。

B.0.2　離散対数問題、DH 問題、DDH 問題

定義 9（離散対数 (Discrete-log (DL)) 問題）　G_q を素数位数 q の巡回群（演算を乗法で書くことにする）とする。元 $g, h \in G_q$ が与えられたとき、$h = g^x$ なる $x \in \mathbb{Z}/q\mathbb{Z}$ を求める問題を G_q 上の離散対数問題という。

$h = g^x$ なる x を、$x = \log_g(h)$ と書く。

定義 10（Diffie-Hellman (DH) 問題）　G_q を素数位数 q の巡回群（演算を乗法で書くことにする）とする。元 $g, h_1, h_2 \in G_q$ が与えられたとき、$k = g^{xy} = h_2^x = h_1^y$ なる $k \in G_q$ を求める問題を G_q 上の DH 問題という。ここで、$x := \log_g(h_1)$, $y := \log_g(h_2)$ とする。

別の言い方をすると、g, $h_1 = g^x$, $h_2 = g^y$ から、x, y は教えられずに g^{xy} を計算する問題を DH 問題という。

定義 11（Decisional Diffie-Hellman (DDH) 問題）　G_q を素数位数 q の巡回群（演算を乗法で書くことにする）とする。G_q の元 $g_1, g_2, h_1, h_2 \in G_q$ に対して、$\log_{g_1}(h_1) = \log_{g_2}(h_2)$ であるとき、DDH 関係を満たしているという。

$$\mathrm{DDH}(G_q^4) := \{(g_1, g_2, h_1, h_2) \in G_q^4 \mid \log_{g_1}(h_1) = \log_{g_2}(h_2)\}.$$

DDH 関係か、否かを判定する問題を G_q 上の DDH 問題という。

G_q の元の 4 つ組の集合 $G_q^4 = \{(g_1, g_2, h_1, h_2) \mid g_1, g_2, h_1, h_2 \in G_q\}$ とすると、$\mathrm{DDH}(G_q^4) \subset G_q^4$.

注釈 12　DDH 関係とは、図 4 のように 4 元のうち 3 元を選んだとき、最後の 1 元が DH 問題の解になっている関係と言い直すことができる。

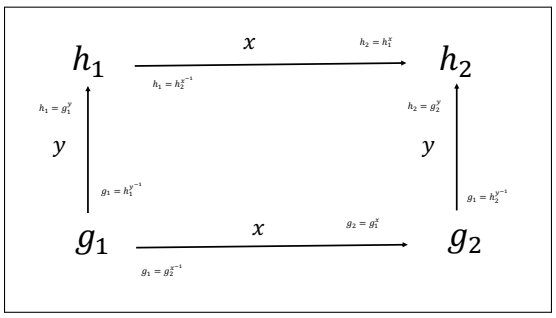

図 4　DDH 関係の図

$\mathsf{KGen}_{\mathsf{group}}$ を次のような確率的アルゴリズムとする。

1. $\mathsf{KGen}_{\mathsf{group}}$ はセキュリティパラメータ 1^κ を入力として受け取る。
2. $|q| = \kappa$ となる素数と、q を位数とする巡回群 G_q を選ぶ。
3. $(G_q, , q)$ を出力する。

このアルゴリズムの試行を

$$(G_q, q) \leftarrow \mathsf{KGen}_{\mathsf{group}}(1^\kappa)$$

と書く。

定義 13（DL 仮定）　DL 問題を解くアルゴリズム A の成功確率を

$$\mathsf{Adv}^{\mathrm{DL}}_{A,\mathsf{KGen}_{\mathsf{group}}}(\kappa) := \Pr[(G_q, q) \leftarrow \mathsf{KGen}_{\mathsf{group}}(1^\kappa); (g, h) \leftarrow G_q^2; x \leftarrow A(g, h) : y = g^x]$$

と定義する。ある巡回群生成アルゴリズム $\mathsf{KGen}_{\mathsf{group}}$ が存在して、いかなる多項式時間アルゴリズムの敵 A に対しても $\mathsf{Adv}^{\mathrm{DL}}_{A,\mathsf{KGen}_{\mathsf{group}}}(\kappa) = \mathsf{negl}(\kappa)$ となるとする。このような $\mathsf{KGen}_{\mathsf{group}}$ の存在を仮定することを DL 仮定と言う。

定義 14（DH 仮定）　DH 問題を解くアルゴリズム A の成功確率を

$$\mathsf{Adv}^{\mathrm{DH}}_{A,\mathsf{KGen}_{\mathsf{group}}}(\kappa) := \Pr[(G_q, q) \leftarrow \mathsf{KGen}_{\mathsf{group}}(1^\kappa); (g, h_1, h_2) \leftarrow G_q^3 : A(g, h_1, h_2) = g^{xy}]$$

と定義する（ただし $x = \log_g(h_1)$, $y = \log_g(h_2)$）。ある巡回群生成アルゴリズム $\mathsf{KGen}_{\mathsf{group}}$ が存在して、いかなる多項式時間アルゴリズムの敵 A に対しても $\mathsf{Adv}^{\mathrm{DH}}_{A,\mathsf{KGen}_{\mathsf{group}}}(\kappa) = \mathsf{negl}(\kappa)$ となるとする。このような $\mathsf{KGen}_{\mathsf{group}}$ の存在を仮定することを DH 仮定と言う。

定義 15（DDH 仮定）　DDH 関係を判定するアルゴリズム D の成功確率を

$$\mathsf{Adv}^{\mathrm{DDH}}_{D,\mathsf{KGen}_{\mathsf{group}}}(\kappa) := |\Pr_{G_q, U_{\mathrm{DDH}}(G_q^4)}[D(U_{\mathrm{DDH}}(G_q^4)) = 1] - \Pr_{G_q, U(G_q^4)}[D(U(G_q^4)) = 1]|$$

と定義する。ただし、G_q は、$\mathsf{KGen}_{\mathsf{group}}(1^\kappa)$ の分布に従った確率変数、$U_{\mathrm{DDH}}(G_q^4)$ は、$\mathrm{DDH}(G_q^4)$ 上の一様分布に従う確率変数、$U(G_q^4)$ は、G_q^4 上の一様分布に従う確率変数である。ある巡回群生成アルゴリズム $\mathsf{KGen}_{\mathsf{group}}$ が存在して、いかなる多項式時間アルゴリズムの敵 D に対しても $\mathsf{Adv}^{\mathrm{DDH}}_{D,\mathsf{KGen}_{\mathsf{group}}}(\kappa) = \mathsf{negl}(\kappa)$ となるとする。このような $\mathsf{KGen}_{\mathsf{group}}$ の存在を仮定することを DDH 仮定と言う。

付録 C　ランダムオラクルモデル (Random Oracle Model)

ランダムオラクルモデルとは、攻撃者を含む全てのアルゴリズムがランダム関数にオラクルアクセスできる環境のことを言う。例えばハッシュ関数 $H : X \to Y$ をランダム関数とみなすとは、H が全ての関数の集合 $\mathsf{Func}(X, Y) = \{f : X \to Y\}$ からランダムに選ばれたものと考えるということである。ハッシュ関数をランダム関数とみなし、そのハッシュ関数へは誰もがオラクルアクセスできるモデルで暗号方式の安全性を証明したとき、その暗号方式は**ランダムオラクルモデルで安全**という。

C.1　公開鍵暗号の場合

公開鍵暗号 Π で使われるハッシュ関数をランダム関数と扱うと、ランダムオラクルモデルでの IND-CCA 安全性ゲームでの敵 A の試行は、次のようになる。

$\mathsf{Expt}^{\mathsf{cca\text{-}ro}}_{\Pi,A}(\kappa)$:
$H \leftarrow \mathsf{Func}$
$(pk, sk) \leftarrow \mathbf{K}^H(1^\kappa)$
$(m_0, m_1, st) \leftarrow A_1^{\mathbf{D}^H_{sk},H}(pk)$
$b^* \leftarrow \{0,1\}; \mathsf{ct}^* \leftarrow \mathbf{E}^H(pk, m_{b^*})$
$b \leftarrow A_2^{\mathbf{D}^H_{sk},H}(st, \mathsf{ct}^*)$
If $b = b^*$,　return 1 else return 0.

A のアドバンテージは次のようになる。

$$\mathsf{Adv}^{\mathsf{cca\text{-}ro}}_{A,\Pi}(\kappa) = \left|2\Pr[\mathsf{Expt}^{\mathsf{cca}}_{\Pi,A}(\kappa) = 1] - 1\right|.$$

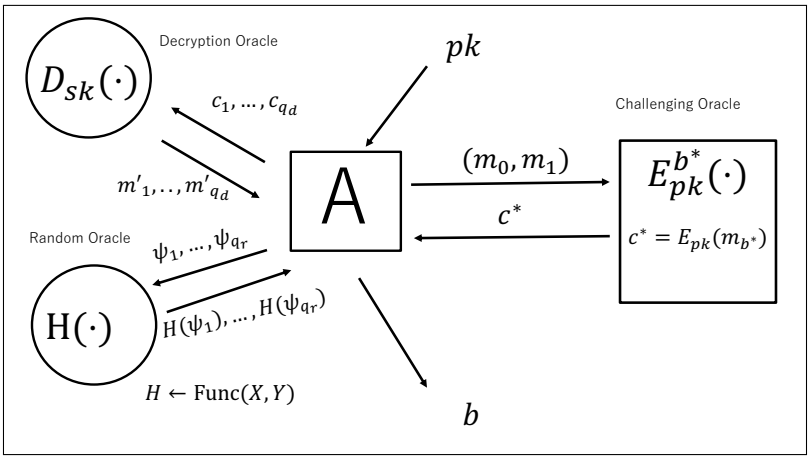

図 5　IND-CCA in ROM