KyberとSaberの耐量子計算機安全性 (概要版)

Varun Maram[1]　草川　恵太[2,a)]

概要：Grubbs, Maram, Paterson (EUROCRYPT 2022) は, Kyber や Saber で採用されている藤崎岡本変換の変種について, 量子ランダムオラクルモデルにおける IND-CCA 安全性の証明がなされていないことを指摘した. Bernstein により Zhandry の量子ランダムオラクルの量子識別不可能性を用いると証明が通るだろうと指摘されたが, 具体的なバウンドは分かっていない. 本稿では, Kyber と Saber の量子ランダムオラクルモデルにおける IND-CCA 安全性（と匿名性）を別の手法を用いて証明し, 具体的な不等式を与える.

キーワード：耐量子計算機暗号, NIST PQC 標準化, KEM, Kyber, Saber.

Post-Quantum Security of Kyber and Saber (Extended Abstract)

Varun Maram[1]　Keita Xagawa[2,a)]

Abstract: Grubbs, Maram, and Paterson (EUROCRYPT 2022) pointed out that Kyber and Saber gave two tweaks for the implicit-rejection version of the Fujisaki-Okamoto (FO) transform and there is no IND-CCA security proof for the tweaked FO transform in the quantum random oracle model. Bernstein suggested using Zhandry's quantum indifferentiability (CRYPTO 2019) to remedy the IND-CCA security proof but there is no concrete bound for the IND-CCA security.
This paper gives explicit security proof for the tweaked FO transform in the quantum random model and applies it to Kyber and Saber. Additionally, we apply the technique to those anonymity.

Keywords: Post-Quantum Cryptography, NIST PQC Standardization, KEM, Kyber, Saber.

1. 導入

米国の標準技術開発局 (NIST, National Institute of Standards and Technology) は耐量子計算機安全な公開鍵暗号・鍵カプセル化方式と署名方式の標準化を行っている. NIST は 2022 年 7 月, 公開鍵暗号・鍵カプセル化方式として Kyber を, 署名方式として Dilithium, Falcon, SPHINCS+の三方式を選定した [ADC+22]. 2024 年までには正式に標準規格が作成される予定であり, 今後世の中に広まっていくことが予想される.

Kyber/Saber の IND-CCA 安全性:

Kyber の仕様書では「藤崎岡本変換により量子ランダムオラクルモデル (QROM) でも IND-CCA 安全性を証明できる [HHK17], [SXY18]」と書いてある. しかし, Grubbs, Maram, Paterson [GMP22] は, Kyber や Saber で採用されている藤崎岡本変換の変種について, QROM での IND-CCA 安全性の証明がなされていないことを指摘した. Bernstein により Zhandry の量子ランダムオラクルの量子識別不可能性を用いると証明が通るだろうと指摘されたが, 具体的なバウンドは分かっていない. つまり 2022 年 8 月現在, Kyber の QROM での IND-CCA 安全性は証明されていない.

貢献:

本稿では, Kyber と Saber の量子ランダムオラクルモデルにおける IND-CCA 安全性（と匿名性）を別の手法を用いて証明し, 具体的な不等式を与える.

[1] ETH Zürich
[2] 日本電信電話株式会社 社会情報研究所
Social Informatics Laboratories, NTT Coporation
[a)] keita.xagawa@ntt.com

2. 準備 1

2.1 量子ランダムオラクルに関する補題

本稿では以下の 4 つの補題を用いる.

補題 1 ([Zha15], Theorem 3.1)**.** ある定数 $C < 648$ が存在し, 以下が言える*:* $\mathcal{X}$ と $\mathcal{Y}$ を有限集合とする. $H\colon \mathcal{X} \to \mathcal{Y}$ をランダムオラクルとする. 無限の能力を持つ量子の敵 $\mathcal{A}$ は H への量子クエリを高々 q 回行うとする. このとき,

$$\Pr[H(x_0) = H(x_1) \wedge x_0 \neq x_1 \mid (x_0, x_1) \leftarrow \mathcal{A}^H] \leq \frac{C(q+1)^3}{|\mathcal{Y}|}$$

が成立する.

補題 2 ([BHH+19], Corollary 1)**.** $\mathcal{K}$, $\mathcal{X}$, $\mathcal{Y}$ を有限集合とする. $H\colon \mathcal{K} \times \mathcal{X} \to \mathcal{Y}$ と $R\colon \mathcal{X} \to \mathcal{Y}$ を独立したランダムオラクルとする. $k \leftarrow \mathcal{K}$ をランダムにとり, $F_0(\cdot) \coloneqq H(k, \cdot)$ と $F_1(\cdot) \coloneqq R(\cdot)$ と定義する. 無限の能力を持つ量子の敵 $\mathcal{A}$ は高々 q 回の量子クエリを行うとする. このとき,

$$|\Pr[1 \leftarrow A^{H,F_0}] - \Pr[1 \leftarrow A^{H,F_1}]| \leq \frac{2q}{\sqrt{|\mathcal{K}|}}.$$

次に一方向性から識別不可能性を導出する, Oneway-to-hiding (OW2H) 補題を紹介する.

補題 3 (Original OW2H [Unr14])**.** $\mathcal{X}$ と $\mathcal{Y}$ を有限集合とし, $H\colon \mathcal{X} \to \mathcal{Y}$ をランダムオラクルとする. H に高々 q 回アクセスするオラクルアルゴリズム A^H を考える. B^H を以下のアルゴリズムとして定義する*:* x を入力とする. $i \leftarrow \{1, \ldots, q\}$ と $y \leftarrow\!\!\$\, \mathcal{Y}$ をランダムに選び, $A^H(x, y)$ を i 番目のクエリの前まで動かし, i 番目のクエリを計算基底で観測し, 観測結果を出力する *(*A が i 未満のクエリしかしない場合は, B は $\perp$ を出力する*)*.

$$P_A^1 = \Pr[1 \leftarrow A^H(x, H(x)) \mid x \leftarrow\!\!\$\, \mathcal{X}]$$
$$P_A^2 = \Pr[1 \leftarrow A^H(x, y) \mid x \leftarrow\!\!\$\, \mathcal{X}, y \leftarrow\!\!\$\, \mathcal{Y}]$$
$$P_B = \Pr[x \leftarrow B^H(x) \mid x \leftarrow\!\!\$\, \mathcal{X}]$$

すると, $|P_A^1 - P_A^2| \leq 2q\sqrt{P_B}$.

補題 4 (一般化 OW2H [AHU19], Theorem 3)**.** $\mathcal{X}$ と $\mathcal{Y}$ を有限集合とする. $\mathcal{S} \subseteq \mathcal{X}$ を確率変数とする. $G, H\colon \mathcal{X} \to \mathcal{Y}$ を関数とし, $x \notin \mathcal{S}$ で, $G(x) = H(x)$ とする. z をビット列とする. *(*$\mathcal{S}, G, H, z$ は任意の同時生成分布を持つとする*)*

H に高々 q 回アクセスするオラクルアルゴリズム A^H を考える. B^H を以下のアルゴリズムとして定義する*:* z を入力とする. $i \leftarrow \{1, \ldots, q\}$ をランダムに選び, $A^H(z)$ を i 番目のクエリの前まで動かし, i 番目のクエリを計算基底で観測し, 観測結果 $\mathcal{T} = \{t_1, \ldots, t_{|\mathcal{T}|}\}$ を出力する *(*パラレルクエリを考えているので複数の結果を得る. A が i 未満のクエリしかしない場合は, B は $\perp$ を出力する*)*.

$$P_{\text{left}} \coloneqq \Pr[1 \leftarrow A^H(z)]$$
$$P_{\text{right}} \coloneqq \Pr[1 \leftarrow A^G(z)]$$
$$P_{\text{guess}} \coloneqq \Pr[\mathcal{S} \cap \mathcal{T} \neq \emptyset : \mathcal{T} \leftarrow B^H(x)]$$

と定義する. すると $|P_{\text{left}} - P_{\text{right}}| \leq 2q\sqrt{P_{\text{guess}}}$ が成立する. P_B の B^H を B^G に置き換えても同じバウンドが得られる.

2.2 公開鍵暗号

公開鍵暗号 PKE は KGen, Enc, Dec の確率的多項式時間アルゴリズムの 3 つ組で定義される.

- KGen: セキュリティパラメータ 1^λ を入力とし, 暗号化鍵 pk と 復号鍵 sk を出力する.
- Enc: 暗号化鍵 pk と平文 m を入力とし, 暗号文 c を出力する.
- Dec: 復号鍵 sk と暗号文 c を入力とし, 平文 m または拒否シンボル $\perp$ を出力する.

単射:

(Taken from [BHH+19].) 決定性の公開鍵暗号 PKE が η-**単射**であるとは,

$$\Pr_{(\mathsf{pk},\mathsf{sk}) \leftarrow \mathsf{KGen},\mathsf{H}}[\mathsf{Enc}(\mathsf{pk}, \cdot) \text{ is not injective}] \leq \eta.$$

を満たすことをいう. $\eta = 0$ の場合は, 単に**単射**という

正確性:

公開鍵暗号 PKE が δ-**正確**であるとは, 任意の m について

$$\Pr[\mathsf{Dec}(\mathsf{sk}, c) \neq m \mid (\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KGen}(), c \leftarrow \mathsf{Enc}(\mathsf{pk}, m)] \leq \delta.$$

を満たすときをいう. また, 公開鍵暗号 PKE が δ-*HHK*-**正確**であるとは,

$$\mathop{\mathrm{Exp}}_{(\mathsf{pk},\mathsf{sk}) \leftarrow \mathsf{KGen}()}[\max_{m \in \mathcal{M}} \Pr[\mathsf{Dec}(\mathsf{sk}, c) \neq m \mid c \leftarrow \mathsf{Enc}(\mathsf{pk}, m)]] \leq \delta.$$

を満たすときをいう.

2.3 鍵カプセル化方式

鍵カプセル化方式 KEM は KGen, Encap, Decap の確率的多項式時間アルゴリズムの 3 つ組で定義される.

- KGen: セキュリティパラメータ 1^λ を入力とし, 暗号化鍵 pk と 復号鍵 sk を出力する.
- Encap: 暗号化鍵 pk を入力とし, 暗号文 c と鍵 k を出力する.
- Decap: 復号鍵 sk と暗号文 c を入力とし, 鍵 k または拒否シンボル $\perp$ を出力する.

KEM の IND-CCA 安全性:

$\mathsf{KEM} = (\mathsf{KGen}, \mathsf{Encap}, \mathsf{Decap})$ に対する敵 $\mathcal{A}$ の IND-CCA ゲーム $\mathsf{G}_{\text{IND-CCA}}$ を図 1 で定義する. 敵 $\mathcal{A}$ の優位性を

$$\mathbf{Adv}^{\text{IND-CCA}}_{\mathsf{KEM}}(\mathcal{A}) \coloneqq |\Pr[\mathsf{G}_{\text{IND-CCA}} = 1] - 1/2|$$

で定義する. 任意の量子多項式時間の敵 $\mathcal{A}$ に対してその優位性 $\mathbf{Adv}^{\text{IND-CCA}}_{\mathsf{KEM}}(\mathcal{A})$ が無視できるとき, KEM は IND-CCA 安全であるという.

3. Kyber の QROM での IND-CCA 安全性

Kyber.KEM は Kyber.PKE に図 2 の $\mathsf{FO}^{\not\perp'}$ 変換を適用して得られる KEM である.

定理 1. Kyber.KEM に対する *IND-CCA* ゲームの敵 $\mathcal{A}$ の復号オラクルへのクエリ回数を q_D 回, ランダムオラクル G, H, H' への量子クエリの回数を q_G, q_H, $q_{H'}$ 回であるとする. このとき, $\overline{\mathsf{Kyber}}$.KEM に対する *IND-CCA* ゲームの敵 $\mathcal{B}$ が存在し,

$$\mathbf{Adv}^{\text{IND-CCA}}_{\mathsf{Kyber.KEM}}(\mathcal{A}) \le \mathbf{Adv}^{\text{IND-CCA}}_{\overline{\mathsf{Kyber}}.\mathsf{KEM}}(\mathcal{C}) + \frac{7q_{H'}+2q_H}{2^{128}} + \frac{2C(q_H+1)^3}{2^{256}}$$

が成立する. ここで $C < 648$ は補題 *1* に出てくる定数である. $\mathcal{C}$ の動作時間は $\mathcal{A}$ の動作時間とほぼ同じである.

3.1 準備

Kyber 本体の IND-CCA 安全性証明を行う前に, 中間段階として Pre-Kyber $\overline{\mathsf{Kyber}}$.KEM を考える. 鍵および乱数の生成の際に $h = H(\mathsf{pk})$ も加える FO 変換 $\mathsf{FO}^{\not\perp'}_{\text{pre}}$ を Kyber.PKE に適用して得られたものである. この構成であれば, 既存のテクニックを用いて $\overline{\mathsf{Kyber}}$.KEM の IND-CCA 安全性が証明できる.

さて, $\mathcal{C}$ を $\overline{\mathsf{Kyber}}$.KEM に対する敵とし, 復号オラクルおよびランダムオラクル G, H, H' へのクエリ回数を q'_D, q'_G, q'_H, $q'_{H'}$ とする. 復号オラクルを少しずつ変化させた図 4 にあるゲーム $\overline{\mathsf{G}}_0$, $\overline{\mathsf{G}}_1$, $\overline{\mathsf{G}}_2$ を考える.

Game $\overline{\mathsf{G}}_0$: このゲームは $\overline{\mathsf{Kyber}}$.KEM に対する IND-CCA ゲームである. よって,

$$\left|\Pr[\overline{\mathsf{G}}_0 = 1] - \frac{1}{2}\right| = \mathbf{Adv}^{\text{IND-CCA}}_{\overline{\mathsf{Kyber}}.\mathsf{KEM}}(\mathcal{C}). \tag{1}$$

Game $\overline{\mathsf{G}}_1$: このゲームでは, 復号オラクルは不正な暗号文に対して $H'(s,c)$ を返すのではなく, 新たなランダムオラクル H'' を使って $H''(c)$ を返答する. 量子ランダムオラクルの疑似ランダム性により (,...) ,

$$|\Pr[\overline{\mathsf{G}}_1 = 1] - \Pr[\overline{\mathsf{G}}_0 = 1]| \le 2q'_{H'} \cdot 2^{-256/2} = \frac{2q'_{H'}}{2^{128}}. \tag{2}$$

を得る.

Game $\overline{\mathsf{G}}_2$: このゲームでは, 新たなランダムオラクル $\overline{H}$ を用意し, 復号オラクルは不正な暗号文に対して $H''(c)$ を返すのではなく, $\overline{H}(H(c))$ を返答する. H'' と $\overline{H}$ の両方に敵は直接アクセスできない点に注意する.

今, 敵が $H(c_1) = H(c_2)$ となる不正な暗号文 $c_1 \neq c_2$ をクエリしない限り, $\overline{\mathsf{G}}_1$ と $\overline{\mathsf{G}}_2$ は同じである. 復号オラクルへのクエリは古典的なため, このような衝突を起こすイベントは古典的に検知でき, H の衝突発見への自明な帰着を構成できる. $\overline{\mathsf{G}}_1$ での H へのクエリ回数は q'_H, $\overline{\mathsf{G}}_2$ での H へのクエリ回数は $q'_H + q'_D$ なので, 高々は $q'_H + q'_D$ と考えてよい. このときの衝突を発見できる確率の上界を用いて,

$$|\Pr[\overline{\mathsf{G}}_2 = 1] - \Pr[\overline{\mathsf{G}}_1 = 1]| \le \frac{C(q'_H + q'_D + 1)^3}{2^{256}}, \tag{3}$$

where C (< 648) is the constant from Lemma 1. を得る.

不等式 (1) – (3) を全て合わせて

$$\left|\Pr[\overline{\mathsf{G}}_2 = 1] - \frac{1}{2}\right| \le \mathbf{Adv}^{\text{IND-CCA}}_{\mathsf{Kyber.KEM}}(\mathcal{C}) + \frac{2q'_{H'}}{2^{128}} + \frac{C(q'_H + q'_D + 1)^3}{2^{256}} \tag{4}$$

を得る.

3.2 証明本体

Proof. $\mathcal{A}$ を Kyber.KEM に対する敵とし, 復号オラクルへのクエリ数を q_D, 量子ランダムオラクル G, H, H' へのクエリ数を q_G, q_H, $q_{H'}$ とする. 図 5 にある G_0–G_8 を考える.

Game G_0: このゲームは Kyber.KEM に対する IND-CCA ゲームの中で, 真の鍵 k^* を敵 $\mathcal{A}$ に渡すゲームである.

Game G_1: このゲームでは 3 行目の $m^* \leftarrow H(m^*)$ を省略する.

オリジナルの OW2H 補題を用いて, G_0 と G_1 の差分を抑えることができる. $x := m^*_0 \leftarrow\!\$\, \{0,1\}^{256}$, $y := m^*_1 \leftarrow\!\$\, \{0,1\}^{256}$ とし, $A^H(m^*_0, H(m^*_0))$ は G_0 をシミュレートし, $A^H(m^*_0, m^*_1)$ は G_1 をシミュレートする.

さて, 一方, $A^H(m^*_0, m^*_1)$ における敵 $\mathcal{A}$ は m^*_0 の情報が得られないため, $P_B = 1/2^{256}$ になる.

$$|\Pr[\mathsf{G}_1 = 1] - \Pr[\mathsf{G}_0 = 1]| \le = 2q_H\sqrt{P_B} = \frac{2q_H}{2^{128}}. \tag{5}$$

Game G_2 このゲームでは, 復号オラクルは新たなランダムオラクル H'' を用意して不正な暗号文 c に対しては $H''(H(c))$ を返答する. 準備で行った $\overline{\mathsf{G}}_0$ から $\overline{\mathsf{G}}_2$ への移行と同様に計算を行うと,

$$|\Pr[\mathsf{G}_2 = 1] - \Pr[\mathsf{G}_1 = 1]| \le \frac{2q_{H'}}{2^{128}} + \frac{C(q_H + q_D + 1)^3}{2^{256}}. \tag{6}$$

を得る.

Game G_3 このゲームでは復号オラクルはまた新たなランダムオラクル $\overline{H}$ を用意し, 不正な暗号文 c に対しては $H''(H(c))$ の代わりに $H'(\overline{H}(H(c)), H(c))$ を返答する.

今回は一般化 OW2H 補題 (補題 4) を用いて, バウンドを得たい. われわれは, G_2 中の $(H''(\cdot), H')$ と G_3 中の $(H'(\overline{H}(\cdot),\cdot), H')$ を識別するアルゴリズムを考えている. こ

Game $\mathsf{G}_{\text{IND-CCA}}$		$\mathsf{oDecap}(\mathsf{sk}, c)$	
1 :	$(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KGen}$	**if** $c = c^*$ **then return** $\perp$	
2 :	$(c^*, k_0^*) \leftarrow \mathsf{Encap}(\mathsf{pk})$	**else return** $\mathsf{Decap}(\mathsf{sk}, c)$	
3 :	$k_1^* \leftarrow\!\$\, \mathcal{K}$		
4 :	$b \leftarrow\!\$\, \{0,1\}$		
5 :	$b' \leftarrow \mathcal{C}^{\mathsf{oDecap}(\mathsf{sk},\cdot)}(\mathsf{pk}, c^*, k_b^*)$		
6 :	**return** $(b' = b)$		

図 1 ゲーム $\mathsf{G}_{\text{IND-CCA}}$.

KGen'		$\mathsf{Encap}(\mathsf{pk})$		$\mathsf{Decap}(\mathsf{sk}', c)$	
1 :	$(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KGen}$	1 :	$m \leftarrow\!\$\, \{0,1\}^{256}$	1 :	Parse $\mathsf{sk}' = (\mathsf{sk}, \mathsf{pk}, h, s)$
2 :	$s \leftarrow\!\$\, \{0,1\}^{256}$	2 :	$m \leftarrow H(m)$	2 :	$m' \leftarrow \mathsf{Dec}(\mathsf{sk}, c)$
3 :	$\mathsf{pk}' \leftarrow (\mathsf{pk}, H(\mathsf{pk}))$	3 :	$h \leftarrow H(pk)$	3 :	$(\overline{k}', r') \leftarrow G(m', h)$
4 :	$\mathsf{sk}' \leftarrow (\mathsf{sk}, \mathsf{pk}', s)$	4 :	$(\overline{k}, r) \leftarrow G(m, h)$	4 :	$c' \leftarrow \mathsf{Enc}(\mathsf{pk}, m'; r')$
5 :	**return** $(\mathsf{pk}, \mathsf{sk}')$	5 :	$c \leftarrow \mathsf{Enc}(\mathsf{pk}, m; r)$	5 :	**if** $c' = c$ **then**
		6 :	$k \leftarrow H'(\overline{k}, H(c))$	6 :	**return** $H'(\overline{k}', H(c))$
		7 :	**return** (c, k)	7 :	**else return** $H'(s, H(c))$

図 2 Kyber や Saber で用いられる FO 変換の亜種 $\mathsf{FO}^{\not\perp'}$. $H, H' \colon \{0,1\}^* \to \{0,1\}^{256}$ と $G \colon \{0,1\}^* \to \{0,1\}^{512}$ はハッシュ関数である.

KGen'		$\mathsf{Encap}(\mathsf{pk})$		$\mathsf{Decap}(\mathsf{sk}', c)$	
1 :	$(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KGen}$	1 :	$m \leftarrow\!\$\, \{0,1\}^{256}$	1 :	Parse $\mathsf{sk}' = (\mathsf{sk}, \mathsf{pk}, h, s)$
2 :	$s \leftarrow\!\$\, \{0,1\}^{256}$	2 :	$h \leftarrow H(\mathsf{pk})$	2 :	$m' \leftarrow \mathsf{Dec}(\mathsf{sk}, c)$
3 :	$\mathsf{pk}' \leftarrow (\mathsf{pk}, H(\mathsf{pk}))$	3 :	$(\overline{k}, r) \leftarrow G(m, h)$	3 :	$(\overline{k}', r') \leftarrow G(m', h)$
4 :	$\mathsf{sk}' \leftarrow (\mathsf{sk}, \mathsf{pk}', s)$	4 :	$c \leftarrow \mathsf{Enc}(\mathsf{pk}, m; r)$	4 :	$c' \leftarrow \mathsf{Enc}(\mathsf{pk}, m'; r')$
5 :	**return** $(\mathsf{pk}, \mathsf{sk}')$	5 :	**return** $(c, \overline{k})$	5 :	**if** $c' = c$ **then**
				6 :	**return** $\overline{k}'$
				7 :	**else return** $H'(s, c)$

図 3 Kyber や Saber の内部で用いる FO 変換の亜種 $\mathsf{FO}^{\not\perp'}_{\text{pre}}$. $H, H' \colon \{0,1\}^* \to \{0,1\}^{256}$ と $G \colon \{0,1\}^* \to \{0,1\}^{512}$ はハッシュ関数である.

Games $\overline{\mathsf{G}}_0$–$\overline{\mathsf{G}}_2$		$\mathsf{Decap}(\mathsf{sk}', c)$	
1 :	$(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KGen}'$	1 :	Parse $\mathsf{sk}' = (\mathsf{sk}, \mathsf{pk}, h, s)$
2 :	$(c^*, \overline{k}_0^*) \leftarrow \mathsf{Encap}(\mathsf{pk})$	2 :	$m' \leftarrow \mathsf{Dec}(\mathsf{sk}, c)$
3 :	$\overline{k}_1^* \leftarrow\!\$\, \{0,1\}^{256}$	3 :	$(\overline{k}', r') \leftarrow G(m', h)$
4 :	$b \leftarrow\!\$\, \{0,1\}$	4 :	$c' \leftarrow \mathsf{Enc}(\mathsf{pk}, m'; r')$
5 :	$b' \leftarrow \mathcal{C}^{G,H,H',\mathsf{Decap}(\mathsf{sk}',\cdot)}(\mathsf{pk}, c^*, \overline{k}_b^*)$	5 :	**if** $c' = c$ **then**
6 :	**return** $(b' = b)$	6 :	**return** $\overline{k}'$
		7 :	**else return** $H'(s, c)$ $\quad /\!/\ \overline{\mathsf{G}}_0$
		8 :	**else return** $H''(c)$ $\quad /\!/\ \overline{\mathsf{G}}_1$
		9 :	**else return** $\overline{H}(H(c))$ $\quad /\!/\ \overline{\mathsf{G}}_2$

図 4 ゲーム $\overline{\mathsf{G}}_0$–$\overline{\mathsf{G}}_2$. $H'' : \{0,1\}^* \to \{0,1\}^{256}$ と $\overline{H} : \{0,1\}^{256} \to \{0,1\}^{256}$ は内部のランダムオラクルであり, $\mathcal{A}$ は直接アクセスできない.

Games G_0–G_8		$\mathsf{Decap}(\mathsf{sk}', c)$	
1 :	$(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KGen}'$	1 :	Parse $\mathsf{sk}' = (\mathsf{sk}, \mathsf{pk}, h, s)$
2 :	$m^* \leftarrow\$ \{0,1\}^{256}$	2 :	$m' \leftarrow \mathsf{Dec}(\mathsf{sk}, c)$
3 :	$m^* \leftarrow H(m^*)$ // $\mathsf{G}_0, \mathsf{G}_8$	3 :	$(\overline{k}', r') \leftarrow G(m', h)$
4 :	$(\overline{k}_0^*, r^*) \leftarrow\$ G(m^*, H(pk))$	4 :	$c' \leftarrow \mathsf{Enc}(\mathsf{pk}, m'; r')$
5 :	$\overline{k}_1^* \leftarrow\$ \{0,1\}^{256}$	5 :	**if** $c' = c$ **then**
6 :	$c^* \leftarrow \mathsf{Enc}(\mathsf{pk}, m^*; r^*)$	6 :	**return** $H'(\overline{k}', H(c))$
7 :	$k^* \leftarrow H'(\overline{k}_0^*, H(c^*))$ // G_0–G_3	7 :	**else**
8 :	$k^* \leftarrow H'(\overline{k}_1^*, H(c^*))$ // G_4	8 :	**return** $H'(s, H(c))$ // G_0–G_1, G_7–G_8
9 :	$k^* \leftarrow\$ \{0,1\}^{256}$ // G_5–G_8	9 :	**return** $H''(H(c))$ // G_2, G_6
10 :	$b' \leftarrow \mathcal{A}^{G,H,H',\mathsf{Decap}(\mathsf{sk}',\cdot)}(\mathsf{pk}, c^*, k^*)$	10 :	**return** $H'(\overline{H}(H(c)), H(c))$ // G_3–G_5
11 :	**return** b'		

図 5 ゲーム G_0–G_8. $H'' : \{0,1\}^* \to \{0,1\}^{256}$ と $\overline{H} : \{0,1\}^{256} \to \{0,1\}^{256}$ は内部のランダムオラクルであり, $\mathcal{A}$ は直接アクセスできない.

の関数を読み替えて, G_2 では (H'', H') を, G_3 では (H'', G') を使うようにしたい. そこで, G' を H' の $(\overline{H}(x), x)$ という形式の入力に関してリプログラミングしたオラクルとして定義する. すなわち,

$$G'(y) = \begin{cases} H''(x) & y = (\overline{H}(x), x) \text{ with } x \in \{0,1\}^{256} \text{ のとき} \\ H'(y) & \text{otherwise.} \end{cases}$$

さて補題 4 を適用する. 今, $\mathcal{S} = \{(\overline{H}(x), x) \mid x \in \{0,1\}^{256}\}$ である. $\Pr[\mathsf{G}_2 = 1] = P_{\text{left}}$, $\Pr[\mathsf{G}_3 = 1] = P_{\text{right}}$ と定義できて, $|\Pr[\mathsf{G}_3 = 1] - \Pr[\mathsf{G}_2 = 1]| \leq 2q_{H'}\sqrt{P_{\text{guess}}}$ を得る. 一方, G_2 中では $\overline{H}(x)$ は敵 $\mathcal{A}$ にはアクセスできない値であるため, H' オラクルへのクエリを観測した値が $(\overline{H}(x), x)$ となる確率は, 高々 $1/2^{256}$ である. すなわち $P_{\text{guess}} \leq 1/2^{256}$ としてよい. よって,

$$|\Pr[\mathsf{G}_3 = 1] - \Pr[\mathsf{G}_2 = 1]| \leq \frac{2q_{H'}}{2^{128}} \tag{7}$$

を得る.

Game G_4: このゲームでは k^* の生成方法を以下のように変更する. $\overline{k}_1^* \leftarrow\$ \{0,1\}^{256}$ をランダムに選び, $k^* \leftarrow H'(\overline{k}_1^*, H(c^*))$ とする.

ここで準備にある $\overline{\mathsf{Kyber}}.\mathsf{KEM}$ の解析を用いる.

$\overline{\mathsf{G}}_2$ をプレイする以下の敵 $\mathcal{C}$ を考える: $\mathcal{C}^{G,H,H'}(\mathsf{pk}, c^*, \overline{k}_b^*)$ は, $k^* \leftarrow H'(\overline{k}_b^*, H(c^*))$ を計算し, $\mathcal{A}$ を, pk, c^*, k^* を入力として起動する. $\mathcal{C}$ は Decap' を用いて $\mathcal{A}$ の復号オラクルをシミュレートする. もし c を復号する場合, $\mathcal{C}$ の復号オラクルに c をクエリし $\overline{k}$ を得て, $k \leftarrow H'(\overline{k}, H(c))$ を $\mathcal{A}$ に返答する.

もし, $b = 0$ であれば, $k^* \leftarrow H'(\overline{k}_0^*, H(c^*))$ であり G_3 を完璧にシミュレートしている. 逆に $b = 1$ であれば, $k^* \leftarrow H'(\overline{k}_1^*, H(c^*))$ であり G_4 を完璧にシミュレートしている. そのため

$$\begin{aligned} &|\Pr[\mathsf{G}_4 = 1] - \Pr[\mathsf{G}_3 = 1]| \\ &= |\Pr[1 \leftarrow \mathcal{C} \mid b = 1] - \Pr[1 \leftarrow \mathcal{C} \mid b = 0]| \\ &= 2 \cdot \left| \Pr[\overline{\mathsf{G}}_2 = 1] - \frac{1}{2} \right| \end{aligned}$$

となる.

元の $\overline{\mathsf{G}}_2$ のバウンドを用いて,

$$\begin{aligned} &|\Pr[\mathsf{G}_4 = 1] - \Pr[\mathsf{G}_3 = 1]| \\ &\leq 2\mathbf{Adv}^{\text{IND-CCA}}_{\overline{\mathsf{Kyber}}.\mathsf{KEM}}(\mathcal{C}) + \frac{4q_{H'}}{2^{128}} + \frac{2C(q_H+1)^3}{2^{256}} \end{aligned} \tag{8}$$

を得る. (クエリ回数の読み替えに注意)

Game G_5: 次に k^* をランダムにする.

これは, 新たなランダムオラクル H_5 を用意して $k^* = H_5(H(c^*))$ と計算したと読み替えてもよい. すると, 補題 2 を用いることができ,

$$|\Pr[\mathsf{G}_5 = 1] - \Pr[\mathsf{G}_4 = 1]| \leq \frac{2q_{H'}}{2^{128}} \tag{9}$$

を得る.

Game G_6: このゲームでは復号オラクルが不正な暗号文 c に対して $H''(H(c))$ を返答する. G_2 から G_3 へのゲームホップの逆と考えてよいので, 同等の議論から,

$$|\Pr[\mathsf{G}_6 = 1] - \Pr[\mathsf{G}_5 = 1]| \leq \frac{2q_{H'}}{2^{128}} \tag{10}$$

を得る.

Game G_7: このゲームでは, 復号オラクルが不正な暗号文 c に対して $H'(s, H(c))$ を返答する. G_1 から G_2 へのゲームホップの逆と考えてよいので, 同等の議論から,

$$|\Pr[\mathsf{G}_7 = 1] - \Pr[\mathsf{G}_6 = 1]| \leq \frac{2q_{H'}}{2^{128}} + \frac{C(q_H+q_D+1)^3}{2^{256}} \tag{11}$$

Game G_8: このゲームでは, $m \leftarrow H(m)$ の行を戻す. G_0 から G_1 へのゲームホップの逆と考えてよいので, 同等の議論から,

$$|\Pr[\mathsf{G}_8 = 1] - \Pr[\mathsf{G}_7 = 1]| \leq \frac{2q_H}{2^{128}} \tag{12}$$

を得る.

G_8 は $\mathsf{Kyber}.\mathsf{KEM}$ に対する IND-CCA ゲームの中で, ランダムな鍵 k^* を敵 $\mathcal{A}$ に渡すゲームである. したがって,

$$2 \cdot \mathbf{Adv}^{\text{IND-CCA}}_{\mathsf{Kyber}.\mathsf{KEM}}(\mathcal{A}) = |\Pr[\mathsf{G}_8 = 1] - \Pr[\mathsf{G}_0 = 1]|$$

を得る.

By collecting the above bounds (5) - (12), we obtain

$$\begin{aligned} \mathbf{Adv}^{\text{IND-CCA}}_{\mathsf{Kyber}.\mathsf{KEM}}(\mathcal{A}) &\leq \mathbf{Adv}^{\text{IND-CCA}}_{\overline{\mathsf{Kyber}}.\mathsf{KEM}}(\mathcal{C}) \\ &\quad + \frac{7q_{H'} + 2q_H}{2^{128}} + \frac{2C(q_H+1)^3}{2^{256}}. \end{aligned} \tag{13}$$

□

項 $\mathbf{Adv}^{\text{IND-CCA}}_{\overline{\mathsf{Kyber}}.\mathsf{KEM}}(\mathcal{C})$ については, 既存の成果を用いればよい.

- [JZC+18] (+[HHK17]): $\mathsf{Kyber}.\mathsf{PKE}$ の IND-CPA 安全性と δ-正確性を仮定する.
- [SXY18]: $\mathsf{Kyber}.\mathsf{PKE}$ の PR-CPA 安全性と δ-正確性を仮定する.
- [BHH+19]+[KSS+20]: $\mathsf{Kyber}.\mathsf{PKE}$ の IND-CPA 安全性, δ-正確性, η-単射性を仮定する.

参考文献

[ADC+22] G. Alagic, D. Apon, D. Cooper, Q. Dang, T. Dang, J. Kelsey, J. Lichtinger, C. Miller, D. Moody, R. Peralta, R. Perlner, A. Robinson, D. Smith-Tone, and Y.-K. Liu. Status Report on the Third Round of the NIST Post-Quantum Cryptography Standardization Process. NISTIR 8413, 2022.

https://doi.org/10.6028/NIST.IR.8413

[AHU18] A. Ambainis, M. Hamburg, and D. Unruh. Quantum security proofs using semi-classical oracles. Cryptology ePrint Archive, Report 2018/904, 2018. https://eprint.iacr.org/2018/904.

[AHU19] A. Ambainis, M. Hamburg, and D. Unruh. Quantum security proofs using semi-classical oracles. In *CRYPTO 2019, Part II*, pages 269–295, 2019.

[BHH+19] N. Bindel, M. Hamburg, K. Hövelmanns, A. Hülsing, and E. Persichetti. Tighter proofs of CCA security in the quantum random oracle model. In *TCC 2019, Part II*, pages 61–90, 2019.

[Dec03] A. W. Dent. A designer's guide to KEMs. In *9th IMA International Conference on Cryptography and Coding*, pages 133–151, 2003.

[DFMS22] J. Don, S. Fehr, C. Majenz, and C. Schaffner. Online-extractability in the quantum random-oracle model. In *EUROCRYPT 2022, Part III*, pages 677–706, 2022.

[GMP22] P. Grubbs, V. Maram, and K. G. Paterson. Anonymous, robust post-quantum public key encryption. In *EUROCRYPT 2022, Part III*, pages 402–432, 2022.

[HHK17] D. Hofheinz, K. Hövelmanns, and E. Kiltz. A modular analysis of the Fujisaki-Okamoto transformation. In *TCC 2017, Part I*, pages 341–371, 2017.

[HHM22] K. Hövelmanns, A. Hülsing, and C. Majenz. Failing gracefully: Decryption failures and the fujisaki-okamoto transform. Cryptology ePrint Archive, Report 2022/365, 2022. https://eprint.iacr.org/2022/365.

[HKSU20] K. Hövelmanns, E. Kiltz, S. Schäge, and D. Unruh. Generic authenticated key exchange in the quantum random oracle model. In *PKC 2020, Part II*, pages 389–422, 2020.

[JZC+18] H. Jiang, Z. Zhang, L. Chen, H. Wang, and Z. Ma. IND-CCA-secure key encapsulation mechanism in the quantum random oracle model, revisited. In *CRYPTO 2018, Part III*, pages 96–125, 2018.

[KKPP20] S. Katsumata, K. Kwiatkowski, F. Pintore, and T. Prest. Scalable ciphertext compression techniques for post-quantum KEMs and their applications. In *ASIACRYPT 2020, Part I*, pages 289–320, 2020.

[KSS+20] V. Kuchta, A. Sakzad, D. Stehlé, R. Steinfeld, and S. Sun. Measure-rewind-measure: Tighter quantum random oracle model proofs for one-way to hiding and CCA security. In *EUROCRYPT 2020, Part III*, pages 703–728, 2020.

[SXY18] T. Saito, K. Xagawa, and T. Yamakawa. Tightly-secure key-encapsulation mechanism in the quantum random oracle model. In *EUROCRYPT 2018, Part III*, pages 520–551, 2018.

[TU16] E. E. Targhi and D. Unruh. Post-quantum security of the Fujisaki-Okamoto and OAEP transforms. In *TCC 2016-B, Part II*, pages 192–216, 2016.

[Unr14] D. Unruh. Revocable quantum timed-release encryption. In *EUROCRYPT 2014*, pages 129–146, 2014.

[Xag22] K. Xagawa. Anonymity of NIST PQC round 3 KEMs. In *EUROCRYPT 2022, Part III*, pages 551–581, 2022.

[Zha15] M. Zhandry. A note on the quantum collision and set equality problems. *Quantum Information and Computation*, 15(7–8), 2015.

[Zha19] M. Zhandry. How to record quantum queries, and applications to quantum indifferentiability. In *CRYPTO 2019, Part II*, pages 239–268, 2019.