秘密鍵の漏洩耐性を有する鍵隔離暗号

淺野 京一[1,a)]　岩本 貢[1]　渡邉 洋平[1,2]

概要：公開鍵暗号において, 秘密鍵の漏洩への対策の 1 つとして鍵隔離暗号がある. 鍵隔離暗号とは, 利用者が秘密鍵として復号鍵と補助鍵を持ち, 補助鍵を用いて復号鍵を定期的に更新することで, ある期間の復号鍵の全体が漏洩しても漏洩していない期間の安全性を保証する暗号である. このように鍵隔離暗号では, 漏洩していない期間の安全性は保証されるが, 部分的にでも漏洩してしまった期間の安全性を保証していない. 復号鍵が部分的に漏洩した場合でも安全性を保証する暗号として漏洩耐性暗号があるが, 鍵隔離暗号のように復号鍵を更新する機構を持っていないため, 鍵隔離暗号が満たす安全性を満たしていない. そこで本稿では, 鍵隔離暗号が満たす安全性と漏洩耐性暗号が満たす安全性の両方を満たす暗号として, 漏洩耐性鍵隔離暗号を新たに提案する. 漏洩耐性を有する ID ベース暗号や漏洩耐性を有する秘密分散法を用いて, 漏洩耐性鍵隔離暗号の構成を 2 種類示す.

キーワード：鍵隔離暗号, 漏洩耐性公開鍵暗号, 漏洩耐性鍵隔離暗号

Key-Insulated Public-Key Encryption with Secret-Key Leakage Resilience

Kyoichi Asano[1,a)]　Mitsugu Iwamoto[1]　Yohei Watanabe[1,2]

Abstract: *Key-insulated encryption* (KIE) is public-key encryption introduced as one of the countermeasures against secret-key leakage. KIE updates decryption keys with an updating key, called a helper key, to guarantee that even if many decryption keys, where each of them corresponds to each time period, are leaked, no useful information is leaked from ciphertexts encrypted during other time periods. However, KIE does not support resilience against partial key leakage during the non-leaked time periods. Although there are several kinds of *leakage-resilient public-key encryption*, which guarantee resilience against such partial information leakage, they cannot support resilience against *entire* information leakage as in KIE. In this work, we introduce leakage-resilient key-insulated encryption (LR-KIE) that satisfies resilience against both partial and entire secret-key leakage. We show two LR-KIE schemes from any leakage-resilient identity-based encryption scheme and/or any leakage-resilient secret sharing scheme.

Keywords: Key-Insulated Encryption, Leakage-Resilient Public-Key Encryption, Leakage-Resilient Key-Insulated Encryption

1. はじめに

公開鍵暗号において, 秘密鍵の漏洩への対策の一つとして鍵隔離暗号 (Key-Insulated Encryption: KIE) がある. 鍵隔離暗号とは, 利用者が秘密鍵として復号鍵と補助鍵を持ち, 補助鍵を用いて復号鍵を定期的に更新することで, ある期間の復号鍵の全体が漏洩しても漏洩していない期間の安全性を保証する暗号である. 鍵隔離暗号では, 補助鍵はプライベートデバイス (PD) に格納され, ネットワークへの接続頻度が低く, 安全な場所で保管されている USB メモリ等に保存されることが想定される.

また, サイドチャネル攻撃等による秘密鍵の部分的な漏

[1] 電気通信大学
The University of Electro-Communications
[2] 産業技術総合研究所
AIST
[a)] k.asano@uec.ac.jp

洩への対策の 1 つとして, 漏洩耐性暗号 (Leakage-Resilient Public-Key Encryption: LR-PKE) [9, 10] がある. 漏洩耐性暗号は, 秘密情報が部分的に一定の割合漏洩したとしても安全であることが保証されている暗号である.

このように, 期間毎の復号鍵を導入することで, ある期間の復号鍵の全体が漏洩しても安全性を保証可能な暗号技術や, 復号鍵が部分的に漏洩しても安全性を保証できる暗号技術が知られている. 秘密鍵を保管するデバイスの紛失, 盗難等による秘密鍵の (全体) 漏洩, 及びサイドチャネル攻撃等による秘密鍵の部分漏洩, どちらも同時に起こり得る脅威であるが, 現状ではどちらかの安全性を満たした暗号技術しか選択することができない.

1.1 本稿の貢献

本稿では, そのような復号鍵の全体漏洩および部分漏洩が同時に起こったとしても安全性を保証可能な公開鍵暗号の実現を目指す. 具体的には, 鍵隔離暗号に漏洩耐性を持たせた漏洩耐性鍵隔離暗号 (Leakage-Resilient Key-Insulated Encryption: LR-KIE) を提案する. 本稿では 2 種類の安全性レベルを考え, それぞれの安全性を満たす LR-KIE の構成法を提案する. 具体的には, 弱い安全性を満たす LR-KIE は Bellare, Palacio [2] らの手法に基づき漏洩耐性 ID ベース暗号 (Leakage-Resilient Identity-Based Encryption: LR-IBE) から構成し, 強い安全性を満たす LR-KIE については, LR-IBE と漏洩耐性秘密分散法 (Leakage-Resilient Secret Sharing: LR-SS) を用いて構成する.

2. 準備

2.1 記法

有限集合 S に対して, S から一様ランダムに要素 s を取り出すことを $s \leftarrow_{\$} S$ と表記する. また, $|s|$ を s のビット長, $|S|$ を S の要素数とする.

2.2 鍵隔離暗号 (KIE)

KIE は Dodis ら [7] により提案された鍵漏洩に耐性を持つ公開鍵暗号である. KIE において利用者は秘密鍵として復号鍵 dk_t と, 鍵を更新するための補助鍵 hk を持つ. ここで, $t \in \mathcal{T}$ は期間を指し, $\mathcal{T}$ は期間空間である. 補助鍵は PD に格納されるなど, 実環境から隔離されて保存されることが想定され, PD にはネットワークへの接続頻度が低く, 安全な場所で保管されている USB メモリ等が挙げられる. KIE における鍵更新は次の流れで行う：まず KeyUp アルゴリズムを用いて補助鍵 hk を用いて任意の期間 $t' \in \mathcal{T}$ における更新情報 $\mathsf{ku}_{t'}$ を生成し, 続いて Upd アルゴリズムを用いて復号鍵 dk_t と $\mathsf{ku}_{t'}$ から期間 t' の復号鍵 $\mathsf{dk}_{t'}$ を生成する. 通常の公開鍵暗号と異なり, 暗号化の際に現在の期間を指定する必要があり, 暗号文の期間と復号鍵の期間が対応している時のみ復号可能である. すなわち, 復号鍵が漏洩したとしても異なる期間に暗号化された暗号文を復号することはできない. 具体的には, KIE が満たすべき安全性として以下の 2 つがある.

(1) 複数の期間の復号鍵が漏洩しても, 他の漏洩していない期間の暗号文は安全である.

(2) 補助鍵が漏洩しても, 復号鍵が 1 つも漏れていなければ暗号文は安全である.

理想的には両方を満たすことを目指すが, (1) のみ満たすような KIE も weak KIE として考える.

モデル. KIE Π は, 5 つのアルゴリズム (Setup, KeyUp, Upd, Enc, Dec) から構成されていて, それぞれ以下のように定義される.

- $\mathsf{Setup}(1^\lambda) \to (\mathsf{pk}, \mathsf{dk}_0, \mathsf{hk})$：セキュリティパラメータ λ を入力に取って, 公開鍵 pk, 初期復号鍵 dk_0, 補助鍵 hk を出力する. pk には, 期間空間 $\mathcal{T}$ と平文空間 $\mathcal{M}$ が含まれる.
- $\mathsf{KeyUp}(\mathsf{pk}, \mathsf{hk}, t) \to \mathsf{ku}_t$：$\mathsf{pk}$, 期間 $t \in \mathcal{T}$, hk を入力に取って, 期間 t での更新情報 ku_t を出力する.
- $\mathsf{Upd}(\mathsf{pk}, \mathsf{dk}_t, \mathsf{ku}_{t'}) \to \mathsf{dk}_{t'}$：$\mathsf{pk}$, dk_t, $\mathsf{ku}_{t'}$ を受け取り, 期間 t' での復号鍵 $\mathsf{dk}_{t'}$ を出力する.
- $\mathsf{Enc}(\mathsf{pk}, t, \mathsf{M}) \to \mathsf{ct}_t$：$\mathsf{pk}$, $t \in \mathcal{T}$, 平文 M を受け取り, 暗号文 ct_i を出力する.
- $\mathsf{Dec}(\mathsf{pk}, \mathsf{dk}_t, \mathsf{ct}_t) \to \mathsf{M}$：$\mathsf{pk}$, dk_t, ct_t を受け取り, 復号した結果 M を出力する.

正当性. 期間 t に対する暗号文 ct_t は, 復号鍵 dk_t で正しく復号されなければならない. 全ての $\lambda \in \mathbb{N}$, $(\mathsf{pk}, \mathsf{dk}_0, \mathsf{hk}) \leftarrow \mathsf{Setup}(1^\lambda)$, $t, t' \in \mathcal{T}$, $\mathsf{M} \in \mathcal{M}$ に対して, $\mathsf{dk}_t \leftarrow \mathsf{Upd}(\mathsf{pk}, \mathsf{dk}_{t'}, \mathsf{KeyUp}(\mathsf{pk}, \mathsf{hk}, t))$, $\mathsf{Dec}(\mathsf{pk}, \mathsf{dk}_t, \mathsf{Enc}(\mathsf{pk}, t, \mathsf{M})) = \mathsf{M}$ が圧倒的な確率で成立する.

安全性. KIE Π に対して, 攻撃者 $\mathcal{A}$ と挑戦者 $\mathcal{C}$ の間のゲームを考える. ゲームはセキュリティパラメータ λ を入力として受け取り, 以下のように進む: $\mathcal{C}$ は $(\mathsf{pk}, \mathsf{dk}_0, \mathsf{hk}) \leftarrow \mathsf{Setup}(1^\lambda)$ を実行して, pk を $\mathcal{A}$ に送る. $\mathcal{A}$ は適応的に次のクエリを行うことができる. ただし, 以下の 2 種類のクエリのうち, どちらか一方のクエリしか許されない.

- 復号鍵生成クエリ: 攻撃者 $\mathcal{A}$ のクエリ $t \in \mathcal{T}$ に対して, 挑戦者 $\mathcal{C}$ は $\mathsf{dk}_t \leftarrow \mathsf{Upd}(\mathsf{pk}, \mathsf{dk}_0, \mathsf{KeyUpd}(\mathsf{pk}, \mathsf{hk}, t))$ を実行して, dk_t を攻撃者 $\mathcal{A}$ に送る.
- 補助鍵公開クエリ: 攻撃者 $\mathcal{A}$ のクエリに対して, 挑戦者 $\mathcal{C}$ は補助鍵 hk を攻撃者に送る.

攻撃者 $\mathcal{A}$ は, 適当なタイミングでチャレンジクエリを 1 回だけ実行することができる.

- チャレンジクエリ: $\mathcal{A}$ からクエリ $(t^*, \mathsf{M}_0, \mathsf{M}_1)$ ($|\mathsf{M}_0| = |\mathsf{M}_1|$) を受け取り $\mathcal{C}$ は $b \leftarrow_{\$} \{0, 1\}$ を用いて, $\mathsf{ct}^*_{t^*} \leftarrow \mathsf{Enc}(\mathsf{pk}, t^*, \mathsf{M}_b)$ を実行して $\mathcal{A}$ に渡す. t^* は秘密鍵生成クエリでクエリされていないものとする.

チャレンジクエリ終了後, $\mathcal{A}$ は補助鍵公開クエリを行っ

ていない場合のみ, 復号鍵生成クエリを適応的に実行可能である. ただし, t^* に対しては復号鍵生成クエリを実行することはできない. 最後に, $\mathcal{A}$ は b の推測値として, $b' \in \{0,1\}$ を出力して, ゲームを終了する.

定義 2.1 (IND-KI-CPA 安全)**.** KIE Π が IND-KI-CPA 安全であるとは, 任意の確率的多項式時間攻撃者 $\mathcal{A}$ に対して, 上記のゲームでの $\mathcal{A}$ の優位性

$$\mathsf{Adv}^{\mathsf{KIE}}_{\Pi,\mathcal{A}}(\lambda) \coloneqq \left|\Pr[b' = b] - \frac{1}{2}\right|$$

が λ に関して negligible であることである.

上記の安全性は, 復号鍵の漏洩を本節の冒頭で述べた安全性要件の (1), (2) を捉えた定義である. (1) のみを捉えた安全性である IND-wKI-CPA 安全であるには, 補助鍵公開クエリを除いたゲームを考え, 定義 2.1と同様に定義した優位性が negligible であればよい.

2.3 漏洩モデル

前節の KIE では, デバイスの盗難や紛失等による漏洩, すなわち鍵一つ一つが丸ごと漏洩する状況を想定していた. これを本稿では**全体漏洩**と呼ぶ. 一方で, サイドチャネル攻撃等によって鍵の情報が部分的に漏洩する状況も存在する. 本稿ではこれを**部分漏洩**と呼ぶ. サイドチャネル攻撃への対策として様々な耐タンパー技術が知られているが, 部分漏洩の完璧な対策は難しい. そこで, 部分漏洩が起き得ることを前提とし, ある程度の部分漏洩に耐性を持つ暗号技術の研究が進められている.

そのような漏洩耐性を有する公開鍵暗号技術において様々な安全性モデルが考えられているが, 以下では, 2006 年に Di Crescenzo, Lipton, Walfish [6] と Dziembowski [8] が独立に提案した Bounded-Retrieval Model (BRM) に注目する. このモデルでは実用的な観点から, 攻撃者が最大 ℓ ビットの秘密鍵の部分情報を得ることができると仮定する. 具体的には, 攻撃者は秘密鍵を引数に取る任意の効率的に計算可能な関数 $f : \{0,1\}^* \to \{0,1\}$ を選択して実行結果を得る, という攻撃を高々 ℓ 回行うことができるものとする. セキュリティパラメータとは別に攻撃者への漏洩量のパラメータ ℓ を用いることで, 秘密鍵のサイズを柔軟に設定することができる. ただし, 単に大きい ℓ を目指すのではなく, 次で定義される漏洩率をできるだけ大きく達成することを目指す.

$$\text{漏洩率} \coloneqq \frac{\text{漏洩量（すなわち } \ell)}{\text{秘密鍵のサイズ}}$$

紙面の都合上割愛するが, BRM において安全であるためには, その他にも公開鍵のサイズや暗号化, 復号の計算量の効率性を併せて評価する必要がある[*1]. また, 公開鍵暗号以外の暗号技術においては, また異なる漏洩モデルを考える必要がある. 例えば秘密分散法の文脈では, ごく最近**適応的漏洩公開モデル** (Adaptive Leakage and Reveal Model) [5] という強い漏洩モデルが提案されている. 本稿では, 次節の漏洩耐性 ID ベース暗号においては BRM で, 次々節の漏洩耐性秘密分散法については適応的漏洩公開モデルで安全性を定式化する. また, 提案する漏洩耐性を有する KIE については, BRM と適応的漏洩公開モデルの両方を基に安全性を定式化する. 詳細は 3節を参照されたい.

[*1] 詳細は [1] を参照されたい.

2.4 漏洩耐性 ID ベース暗号 (LR-IBE)

IBE とは, 1984 年に Shamir [11] により提案された方式であり, 任意の文字列を公開鍵として扱うことが可能な公開鍵暗号である. 効率的な構成は Boneh, Franklin [4] によって提案された. 従来の公開鍵暗号では, 公開鍵がランダムな値であり, その値をもって公開鍵の持ち主を特定することはできないため, 公開鍵認証基盤による公開鍵証明書が必要となる. 一方, IBE ではメールアドレスや電話番号, 学籍番号等の個人と結び付けられた値 (Identity: ID) を公開鍵とすることが可能なので, 公開鍵証明書を必要としない. IBE では, 任意の文字列から秘密鍵を生成できることから, 秘密鍵生成局 (Private Key Generation: PKG) がマスター秘密鍵を用いて ID に対応する秘密鍵を発行する. この秘密鍵を用いて, 対応する ID で暗号化された暗号文を復号することができる.

モデル. IBE Σ は 4 つのアルゴリズム ($\mathsf{IBE.Init}$, $\mathsf{IBE.GenSK}$, $\mathsf{IBE.Enc}$, $\mathsf{IBE.Dec}$) で構成されており, それぞれ以下のように定義される.

- $\mathsf{IBE.Init}(1^\lambda) \to (\mathsf{mpk}, \mathsf{msk})$： セキュリティパラメータ λ を受け取って, マスター公開鍵, マスター秘密鍵を出力する. 平文空間 $\mathcal{M}$, ID 空間 $\mathcal{ID}$, 秘密鍵空間 $\mathcal{SK}$ はセキュリティパラメータ 1^λ によって定まり, それらは mpk に含まれる.
- $\mathsf{IBE.GenSK}(\mathsf{mpk}, \mathsf{msk}, id) \to \mathsf{sk}_{id}$： mpk と msk, id を受け取り, id に対応する秘密鍵 sk_{id} を出力する.
- $\mathsf{IBE.Enc}(\mathsf{mpk}, id, \mathsf{M}) \to \mathsf{ct}_{id}$： mpk と id, 平文 M を受け取り, 暗号文 ct_{id} を出力する.
- $\mathsf{IBE.Dec}(\mathsf{mpk}, \mathsf{sk}_{id}, \mathsf{ct}_{id}) \to \mathsf{M}$： mpk と sk_{id}, 暗号文 ct_{id} を受け取り, 復号した結果 M を出力する.

正当性. id に対する暗号文 ct_{id} は, 秘密鍵 sk_{id} で正しく復号されなければならない. 全ての $\lambda \in \mathbb{N}$, $(\mathsf{mpk}, \mathsf{msk}) \leftarrow \mathsf{IBE.Init}(1^\lambda)$, $id \in \mathcal{ID}$, $\mathsf{M} \in \mathcal{M}$ に対して, $\mathsf{sk}_{id} \leftarrow \mathsf{IBE.GenSK}(\mathsf{mpk}, \mathsf{msk}, id)$, $\mathsf{IBE.Dec}(\mathsf{mpk}, \mathsf{sk}_{id}, \mathsf{IBE.Enc}(\mathsf{mpk}, id, \mathsf{M})) = \mathsf{M}$ が圧倒的な確率で成立する.

安全性. IBE における安全性について, 漏洩耐性を捉えた定義 [1] を記す. IBE Π に対して, 攻撃者 $\mathcal{A}$ と挑戦者 $\mathcal{C}$ の間のゲームを考える. ゲームはセキュリティパラメータ λ を入力として受け取り, 以下のように進む: $\mathcal{C}$ は

$(\mathsf{mpk}, \mathsf{msk}) \leftarrow \mathsf{IBE.Init}(1^\lambda)$ を実行して, mpk を $\mathcal{A}$ に送る. $\mathcal{A}$ は適応的に次のクエリを行うことができる.

- 秘密鍵生成クエリ: 攻撃者 $\mathcal{A}$ のクエリ $id \in \mathcal{ID}$ に対して, 挑戦者 $\mathcal{C}$ は $\mathsf{sk}_{id} \leftarrow \mathsf{IBE.GenSK}(\mathsf{mpk}, \mathsf{msk}, id)$ を実行して, sk_{id} を攻撃者 $\mathcal{A}$ に送る.
- 秘密鍵漏洩クエリ: 攻撃者 $\mathcal{A}$ のクエリ (id, f) に対して, 挑戦者 $\mathcal{C}$ は $f(\mathsf{sk}_{id})$ を計算して攻撃者 $\mathcal{A}$ に送る. このクエリは一回だけ行うことができる. $f : \mathcal{SK} \to \{0,1\}^\ell$ は効率的に計算可能な関数である.

適応的にクエリを実行したあとに, チャレンジクエリを 1 回だけ実行することができる.

- チャレンジクエリ: $\mathcal{A}$ からクエリ $(id^*, \mathsf{M}_0, \mathsf{M}_1)$ $(|\mathsf{M}_0| = |\mathsf{M}_1|)$ を受け取り $\mathcal{C}$ は $b \leftarrow_\$ \{0,1\}$ を用いて, $\mathsf{ct}^*_{id} \leftarrow \mathsf{IBE.Enc}(\mathsf{mpk}, id^*, \mathsf{M}_b)$ を実行して $\mathcal{A}$ に渡す. ただし, id^* は秘密鍵生成クエリでクエリされていないものとする.

チャレンジクエリ終了後, $\mathcal{A}$ は秘密鍵生成クエリを適応的に実行可能である. ただし, id^* に対しては秘密鍵生成クエリを実行することはできない.

最後に, $\mathcal{A}$ は b の推測値として, $b' \in \{0,1\}$ を出力して, ゲームを終了する.

定義 2.2 (ℓ-IND-ID-lrCPA 安全)**.** IBE Π が ℓ-IND-ID-lrCPA 安全であるとは, 任意の確率的多項式時間攻撃者 $\mathcal{A}$ に対して, 上記のゲームにおいて最大 ℓ 回の秘密鍵漏洩クエリを行う $\mathcal{A}$ の優位性

$$\mathsf{Adv}^{\mathsf{IBE}}_{\Pi,\mathcal{A}}(\lambda) \coloneqq \left|\Pr[b' = b] - \frac{1}{2}\right|$$

が λ に関して negligible であることである.

ℓ-IND-ID-lrCPA 安全である IBE を LR-IBE と呼ぶ. また, 上記のゲームにおいて秘密鍵漏洩クエリを除いた場合, 通常の IBE の安全性である IND-ID-CPA 安全性のゲームになる.

2.5 漏洩耐性秘密分散法 (LR-SS)

秘密分散法 (Secret Sharing: SS) は, 1979 年に Shamir [12] と Blakley [3] によって独立に提案された秘密情報を分散する手法である. 生成されるシェアの個数を n, 復元可能なしきい値を k である秘密分散法を (k,n)-しきい値分散法と呼ぶ.

モデル. $\mathcal{S}$ をシークレット空間, $\mathcal{SH}$ をシェア空間とする. (k,n)-しきい値分散法は 2 つのアルゴリズム (SS.Share, SS.Rec) で構成されていて, それぞれ以下のように定義される.

- $\mathsf{SS.Share}(\mathsf{S}) \to (s_1, \ldots, s_n)$: $\mathsf{S} \in \mathcal{S}$ を入力とし, n 個のシェア $(s_1, \ldots, s_n) \in \mathcal{SH}^n$ を出力する.
- $\mathsf{SS.Rec}(s_{p_1}, \ldots, s_{p_x}) \to \mathsf{S}$: x 個 $(x \geq k)$ のシェア $s_{p_1}, \ldots, s_{p_x}$ を入力とし, S を出力する.

正当性. 全ての $\mathsf{S} \in \mathcal{M}$, $P \in 2^{\{1,\ldots,n\}}$ $(|P| \geq t)$ に対して以下が成り立つ.

$$\Pr[\mathsf{SS.Rec}(\mathsf{SS.Share}(\mathsf{S})_P) = \mathsf{S}] = 1$$

$\mathsf{SS.Share}(\mathsf{S})_P$ は, $(s_1, \ldots, s_n) \leftarrow \mathsf{SS.Share}(\mathsf{S})$ に対して, $\mathsf{SS.Share}(\mathsf{S})_P \coloneqq \{s_i \mid i \in P\}$ である.

安全性. 秘密分散法において, シェアの部分漏洩耐性を有するものを総称して漏洩耐性秘密分散法 (Leakage-Resilient SS: LR-SS) と呼ぶ. Chandran ら [5] は, 新たな漏洩モデルとして適応的漏洩公開モデルを提案しており, これは (k,n)-しきい値分散法において, 全体漏洩を許す $k-1$ 個のシェア以外の $n-(k-1)$ 個のシェアがそれぞれ最大 ℓ ビット部分漏洩する場合の安全性をとらえている. この適応的漏洩公開モデルにおける安全性定義を基に, 以下のように安全性を定義する.

(k,n)-しきい値分散法 Γ に対して, 挑戦者 $\mathcal{C}$ と攻撃者 $\mathcal{D}$ の間のゲームを考える.

攻撃者 $\mathcal{D}$ は $\mathsf{S}_0, \mathsf{S}_1 \in \mathcal{S}$ を取り, 挑戦者 $\mathcal{C}$ に送る. 挑戦者 $\mathcal{C}$ は $\beta \leftarrow_\$ \{0,1\}$ を取り, $(s_1, \ldots, s_n) \leftarrow \mathsf{SS.Share}(\mathsf{S}_\beta)$ を実行する.

攻撃者 $\mathcal{D}$ は, 以下のクエリを最大 φ 回行うことができる. 挑戦者 $\mathcal{C}$ はシェア漏洩クエリで聞かれたインデックスを管理するための空集合 $\mathcal{L}$ を用意する.

- シェア漏洩クエリ：攻撃者 $\mathcal{D}$ はインデックス i と任意の漏洩関数 $g : \mathcal{SH} \to \{0,1\}^\ell$ を挑戦者 $\mathcal{C}$ にクエリし, $\mathcal{C}$ は $i \notin \mathcal{L}$ であるならば $g(s_i)$ を実行し, 結果を $\mathcal{D}$ に送る. インデックス i を $\mathcal{L}$ に追加する.

シェア漏洩クエリの後, 攻撃者 $\mathcal{D}$ は適応的に最大 $k-1$ 回のシェア公開クエリを行える.

- シェア公開クエリ：攻撃者 $\mathcal{D}$ はインデックス $j \in \{1, \ldots, n\} \setminus \mathcal{L}$ をクエリし, $\mathcal{C}$ は s_j を $\mathcal{D}$ に送る.

最後に, $\mathcal{D}$ は β の推測として, $\beta' \in \{0,1\}$ を出力して, ゲームを終了する. $\beta' = \beta$ であるときに, $\mathcal{D}$ の勝利となる.

このときの $\mathcal{D}$ の優位性を以下のように定義する.

$$\mathsf{Adv}^{\mathsf{LR\text{-}SS}}_{\Gamma,\mathcal{D},\ell,\varphi}(\lambda) \coloneqq \left|\Pr[\beta' = \beta] - \frac{1}{2}\right|$$

定義 2.3 ($(\ell, \varphi, \varepsilon_{\mathsf{SS}})$-IND-lrAP 安全)**.** (k,n)-しきい値分散法が $(\ell, \varphi, \varepsilon_{\mathsf{SS}})$-lrAP 安全であるとは, 任意の無制限の計算能力をもつ $\mathcal{D}$ に対して, 上記のゲームでの優位性が $\varepsilon_{\mathsf{SS}}$ 以下になることである.

上記の定義は情報理論的安全性における定義であるが, 対象とする攻撃者 $\mathcal{D}$ を確率的多項式時間攻撃者であるとすれば, 以下のように計算量的安全性における定義となる.

定義 2.4 ((ℓ, φ)-IND-lrAP 安全)**.** (k,n)-しきい値分散法が (ℓ, φ)-IND-lrAP 安全であるとは, 任意の確率的多項式時間攻撃者 $\mathcal{D}$ に対して, 上記のゲームでの $\mathcal{D}$ の優位性が λ に関して negligible であることである.

[5] において, 定義 2.3を満たす構成が提案されている.

3. 本研究の提案

本節では, KIE に部分漏洩耐性をもたせた LR-KIE を提案し, より強い鍵漏洩耐性を有する公開鍵暗号の実現を目指す.

3.1 漏洩耐性鍵隔離暗号 (LR-KIE)

本節では, 全体漏洩だけではなく部分漏洩にも耐性があるような KIE を LR-KIE として定式化する. 従って, モデルは 2.2節で定義したものと同様である. 本稿では, 復号鍵の全体漏洩および部分漏洩, また補助鍵の全体漏洩に対する耐性を考える. KIE における部分漏洩耐性を考えた場合, 特に復号鍵が全体漏洩する場合に攻撃者が得られる情報を整理すると, 攻撃者はチャレンジクエリで用いる期間 t^* 以外の期間の復号鍵を全て得ることができるため, 部分漏洩を考慮する復号鍵は期間 t^* のもののみで十分である. 従って, 本稿では最も単純かつ最低限必要だと考えられる設定として, 任意の期間の復号鍵が 1 つ, 高々 ℓ ビット漏洩する状況を考える. なお, 今回は補助鍵の部分漏洩を対象としない. その妥当性については本節の最後に議論する. ここで, KIE の安全性 (定義 2.1) 同様, 攻撃者は任意の漏洩情報を得られるのではなく, ある程度の制限の中で自由に漏洩情報を得られるようにする必要がある. 本稿では, BRM [6, 8] 及び適応的漏洩公開モデル [5] を基に, 攻撃者に許すクエリの組み合わせを決める. 具体的には, 攻撃者は, 復号鍵生成クエリ, 復号鍵漏洩クエリ, 補助鍵公開クエリの 3 つのクエリからなる以下 2 つの組み合わせを選択して攻撃を行うことができるものとする.

(1') 復号鍵生成クエリと復号鍵漏洩クエリを順番関係なく適応的に行う. すなわち, 復号鍵に関して全体漏洩と部分漏洩の両方が起きた場合であっても安全である. 復号鍵生成クエリでの利用が制限されている期間 t であっても復号鍵漏洩クエリには利用できることに留意されたい.

(2') 復号鍵漏洩クエリを適応的に行った後に, 補助鍵公開クエリを行う. すなわち, 復号鍵の部分漏洩及び補助鍵の全体漏洩が起きた場合であっても安全である.

上記 (1'), (2') は KIE における (1), (2) の復号鍵の部分漏洩を考えた際の自然な拡張となっていることがわかる. (2') において補助鍵公開クエリ後の復号鍵漏洩クエリが制限されているが, これは自明な攻撃を避けるためである. 詳細は本節の最後に議論する.

安全性. KIE Π に対して, 攻撃者 $\mathcal{A}$ と挑戦者 $\mathcal{C}$ の間のゲームを考える. ゲームはセキュリティパラメーター λ を入力として受け取り, ゲームは以下のように進む : $\mathcal{C}$ は $(\mathsf{pk}, \mathsf{dk}_0, \mathsf{hk}) \leftarrow \mathsf{Setup}(1^\lambda)$ を実行して, pk を $\mathcal{A}$ に送る. $\mathcal{A}$ は適応的に次のクエリを行うことができる.

- 復号鍵生成クエリ : 攻撃者 $\mathcal{A}$ のクエリ $t \in \mathcal{T}$ に対して, 挑戦者 $\mathcal{C}$ は $\mathsf{dk}_t \leftarrow \mathsf{Upd}(\mathsf{pk}, \mathsf{KeyUp}(\mathsf{pk}, \mathsf{hk}, t), \mathsf{dk}_0)$ を実行して, dk_t を攻撃者 $\mathcal{A}$ に送る. 攻撃者 $\mathcal{A}$ が補助鍵公開クエリを行っていない場合のみ可能である.
- 復号鍵漏洩クエリ : 攻撃者 $\mathcal{A}$ のクエリ (t, f) に対して, 挑戦者 $\mathcal{C}$ は $f(\mathsf{dk}_t)$ を計算して攻撃者 $\mathcal{A}$ に送る. $f : \mathcal{DK} \to \{0,1\}^\ell$ は効率的に計算可能な関数である. このクエリは補助鍵公開クエリを行っていない場合に 1 回のみ可能である.
- 補助鍵公開クエリ : 挑戦者 $\mathcal{C}$ は補助鍵 hk を攻撃者に送る. 攻撃者 $\mathcal{A}$ が復号鍵生成クエリを行っていない場合のみ可能である.

適応的にクエリを実行したあとに, チャレンジクエリを 1 回だけ実行することができる.

- チャレンジクエリ : $\mathcal{A}$ からクエリ $(t^*, \mathsf{M}_0, \mathsf{M}_1)$ ($|\mathsf{M}_0| = |\mathsf{M}_1|$) を受け取り, $\mathcal{C}$ は $b \leftarrow_\$ \{0,1\}$ を求めて, $\mathsf{ct}_t^* \leftarrow \mathsf{Enc}(\mathsf{pk}, t^*, \mathsf{M}_b)$ を実行して $\mathcal{A}$ に渡す. ただし, t^* は復号鍵生成クエリでクエリされていないとする.

チャレンジクエリ終了後, $\mathcal{A}$ は補助鍵公開クエリを行っていない場合のみ復号鍵生成クエリを適応的に実行可能である. ただし, t^* に対しては復号鍵生成クエリを実行することはできない. 最後に, $\mathcal{A}$ は b の推測値として, $b' \in \{0,1\}$ を出力して, ゲームを終了する.

定義 3.1 (ℓ-IND-KI-lrCPA 安全)**.** KIE Π が ℓ-IND-KI-lrCPA 安全であるとは, 任意の確率的多項式時間攻撃者 $\mathcal{A}$ に対して, 上記のゲームにおいて最大 ℓ ビットの秘密鍵漏洩クエリを行う $\mathcal{A}$ の優位性

$$\mathsf{Adv}^{\mathsf{LR\text{-}KIE}}_{\Pi,\mathcal{A},\ell}(\lambda) := \left| \Pr[b' = b] - \frac{1}{2} \right|$$

が λ に関して negligible であることである.

ℓ-IND-KI-lrCPA 安全である KIE を LR-KIE と呼ぶ. また, 上記のゲームから補助鍵公開クエリを除くことで, ℓ-IND-wKI-lrCPA 安全性を定義できる. ℓ-IND-wKI-lrCPA 安全である KIE を weak LR-KIE と呼ぶ.

上記定義における制限について. 本節冒頭で述べた通り, 上記安全性定義には次の 2 つの制限を設けている.

- 補助鍵の部分漏洩を対象としていない.
- 攻撃者は補助鍵公開クエリ後に復号鍵漏洩クエリを行うことができない.

ここで, それら制限の妥当性について議論する.

まず一つ目の制限について, なぜ補助鍵の部分漏洩を考えないかについて説明する. KIE のユースケースとして, 補助鍵はネットワークへの接続頻度が低く, 安全な場所で保管されている USB メモリ等が想定されている. 従って, 紛失等による全体漏洩は起こり得る一方で, サイドチャネル攻撃等による部分漏洩が起こる状況はほとんどないと考えられる. また, 補助鍵の部分漏洩を考えることでゲーム

の記述および構成が複雑になることも予想されるため, 実用性と定義のシンプルさ, また実現可能性（構成の効率性）の観点から, 本稿では補助鍵の部分漏洩を考えていない.

次に二つ目の制限について, なぜ補助鍵公開クエリの後に復号鍵漏洩クエリができない定義になっているかについて説明する. 補助鍵公開クエリの後の復号鍵漏洩クエリを考えると, 復号鍵漏洩クエリ (t, f) 中の関数 f に補助鍵を埋め込むことができ, KeyUp, Upd の手順を追うことで, 任意の期間の復号鍵を生成し, 漏洩させることが可能である. 具体的には, $f(\cdot)$ に $\mathsf{Upd}(\mathsf{pk}, \cdot, \mathsf{KeyUp}(\mathsf{pk}, \mathsf{hk}, t'))$ を埋め込むことで任意の期間 t' の復号鍵 $\mathsf{dk}_{t'}$ を生成させ, その部分情報を得ることが可能である. 従って, 上記定義で 1 つの期間の復号鍵が高々 ℓ ビット漏洩する状況をとらえるためには, 補助鍵公開クエリ後の復号鍵漏洩クエリを制限する必要がある.

3.2 弱い安全性を満たす構成と安全性証明

IBE から weak KIE を構成する Bellare, Palacio [2] の手法を基に, LR-IBE から weak LR-KIE を構成する.

LR-IBE $\Pi = (\mathsf{IBE.Init}, \mathsf{IBE.KeyGen}, \mathsf{IBE.Enc}, \mathsf{IBE.Dec})$ を用いて LR-KIE $\Sigma = (\mathsf{Setup}, \mathsf{KeyUp}, \mathsf{Upd}, \mathsf{Enc}, \mathsf{Dec})$ を以下のように構成できる.

- $\mathsf{Setup}(1^\lambda)$: セキュリティパラメータ λ を入力にとって, $(\mathsf{mpk}, \mathsf{msk}) \leftarrow \mathsf{IBE.Init}(1^\lambda)$, $\mathsf{dk}_0 \leftarrow \mathsf{IBE.KeyGen}(\mathsf{mpk}, \mathsf{msk}, t_0)$ を実行し, $\mathsf{pk} := \mathsf{mpk}$, $\mathsf{sk}_0 := \mathsf{dk}_0$, $\mathsf{hk} := \mathsf{msk}$ を出力する. pk には, 期間空間 $\mathcal{T}$ と平文空間 $\mathcal{M}$ が含まれている.
- $\mathsf{KeyUp}(\mathsf{pk}, t, \mathsf{hk}) \to \mathsf{ku}_t$: pk と t, hk を受け取り, $\mathsf{ku}_t := (t, \mathsf{hk})$ を出力する.
- $\mathsf{Upd}(\mathsf{pk}, \mathsf{sk}_t, \mathsf{ku}_{t'}) \to \mathsf{sk}_{t'}$: pk, sk_t, $\mathsf{ku}_{t'}$ を入力に取って, $\mathsf{pk} = \mathsf{mpk}$, $\mathsf{ku}_{t'} = (t', \mathsf{hk})$, $\mathsf{hk} = \mathsf{msk}$ とパースし, $\mathsf{sk}_{t'} \leftarrow \mathsf{IBE.KeyGen}(\mathsf{mpk}, \mathsf{msk}, t')$ を出力する.
- $\mathsf{Enc}(\mathsf{pk}, t, \mathsf{M}) \to \mathsf{ct}_t$: $\mathsf{pk} = \mathsf{mpk}$ とパースする. $\mathsf{ct}_t \leftarrow \mathsf{IBE.Enc}(\mathsf{mpk}, t, \mathsf{M})$ を出力する.
- $\mathsf{Dec}(\mathsf{pk}, \mathsf{sk}_t, \mathsf{ct}_t) \to \mathsf{M}$ or $\perp$: $\mathsf{pk} = \mathsf{mpk}$, $\mathsf{sk}_t = \mathsf{dk}_t$ とパースする. $\mathsf{M} \leftarrow \mathsf{IBE.Dec}(\mathsf{mpk}, \mathsf{dk}_t, \mathsf{ct}_t)$ を出力する.

定理 3.2. *IBE が ℓ-IND-ID-lrCPA 安全であるならば, それを用いて構成された KIE は ℓ-IND-wKI-lrCPA 安全である.*

証明. IND-wKI-lrCPA ゲームの攻撃者 $\mathcal{A}$ を用いて, LR-IBE Π の IND-ID-lrCPA 安全性を破る攻撃者 $\mathcal{B}$ を構成する. $\mathcal{C}$ は $(\mathsf{mpk}, \mathsf{msk}) \leftarrow \mathsf{IBE.Init}(1^\lambda)$ を実行して, mpk を $\mathcal{B}$ に送る. $\mathcal{B}$ は $\mathsf{pk} := \mathsf{mpk}$ を $\mathcal{A}$ に渡して, $\mathcal{B}$ が挑戦者として IND-ID-lrCPA ゲームを開始する. $\mathcal{B}$ は以下のようにして, $\mathcal{A}$ からのクエリに対応可能である.

- 復号鍵生成クエリ: $\mathcal{A}$ からのクエリ t に対して, $\mathcal{B}$ は復号鍵生成クエリ t を $\mathcal{C}$ に対して行い, 結果を $\mathcal{A}$ にわたす.
- 復号鍵漏洩クエリ: $\mathcal{A}$ からのクエリ (t, f) に対して, $\mathcal{B}$ は, f の出力の各ビットに対応した関数 $g_1, \ldots, g_\ell$ を作り, $\mathcal{B}$ は秘密鍵漏洩クエリ (t, g_i) $(i \in \{1, \ldots, \ell\})$ を $\mathcal{C}$ に対して行い, 結果を連結したものを $\mathcal{A}$ に返す.

$\mathcal{A}$ のチャレンジクエリ $(t^*, \mathsf{M}_0, \mathsf{M}_1)$ に対して, $\mathcal{B}$ は $\mathcal{C}$ に対してチャレンジクエリ $(t^*, \mathsf{M}_0, \mathsf{M}_1)$ を投げる. $\mathcal{C}$ は $b \leftarrow_\$ \{0, 1\}$ を取り, $\mathsf{ct}_t \leftarrow \mathsf{IBE.Enc}(\mathsf{mpk}, t, \mathsf{M}_b)$ を実行し ct_t を $\mathcal{B}$ に渡す. $\mathcal{B}$ は受け取った ct_t を $\mathcal{A}$ にわたす. この後の $\mathcal{A}$ からのクエリも同様にして $\mathcal{B}$ は対応することができる. $\mathcal{A}$ は推測値 b' を出力してゲームを終了し, $\mathcal{B}$ も推測値 b' を出力してゲームを終了する.

LR-KIE の ℓ-IND-ID-lrCPA ゲームにおける攻撃者 $\mathcal{A}$ に対して $\mathcal{B}$ が挑戦者 $\mathcal{C}$ の挙動を完全に模倣できているため, $\mathcal{B}$ の優位性は,

$$\mathsf{Adv}^{\mathsf{LR\text{-}IBE}}_{\Pi,\mathcal{B},\ell}(\lambda) = \left| \Pr[b = b'] - \frac{1}{2} \right| = \mathsf{Adv}^{\mathsf{LR\text{-}KIE}}_{\Sigma,\mathcal{A},\ell}(\lambda)$$

となる. 従って, LR-IBE が ℓ-IND-ID-lrCPA 安全であるならば, それを用いて構成された KIE は ℓ-IND-wKI-lrCPA 安全である. □

3.3 強い安全性を満たす構成と安全性証明

LR-IBE $\Pi = (\mathsf{IBE.Init}, \mathsf{IBE.KeyGen}, \mathsf{IBE.Enc}, \mathsf{IBE.Dec})$ と 2-out-of-2 LR-SS $\Gamma = (\mathsf{SS.Share}, \mathsf{SS.Rec})$ を用いて LR-KIE $\Sigma = (\mathsf{Setup}, \mathsf{KeyUp}, \mathsf{Upd}, \mathsf{Enc}, \mathsf{Dec})$ を以下のように構成できる.

- $\mathsf{Setup}(1^\lambda) \to (\mathsf{pk}, \mathsf{dk}_0, \mathsf{hk})$: セキュリティパラメータ λ を入力に取って, $(\mathsf{mpk}, \mathsf{msk}) \leftarrow \mathsf{IBE.Init}(1^\lambda)$, $\mathsf{dk}'_0 \leftarrow \mathsf{IBE.KeyGen}(\mathsf{mpk}, \mathsf{msk}, t_0)$, $(s_1, s_2) \leftarrow \mathsf{SS.Share}(\mathsf{msk})$ を実行し, $\mathsf{pk} := \mathsf{mpk}$, $\mathsf{dk}_0 := (\mathsf{dk}'_0, s_1)$, $\mathsf{hk} := s_2$ を出力する. pk には, 期間空間 $\mathcal{T}$ と平文空間 $\mathcal{M}$ が含まれている.
- $\mathsf{KeyUp}(\mathsf{pk}, t, \mathsf{hk}) \to \mathsf{ku}_t$: pk, 期間 t, hk を入力に取って, $\mathsf{ku}_t := (t, \mathsf{hk})$ を出力する.
- $\mathsf{Upd}(\mathsf{pk}, \mathsf{dk}_t, \mathsf{ku}_{t'}) \to \mathsf{dk}_{t'}$: pk, dk_t, $\mathsf{ku}_{t'}$ を入力に取って, $\mathsf{pk} = \mathsf{mpk}$, $\mathsf{dk}_t = (\mathsf{dk}'_t, s_1)$, $\mathsf{ku}_{t'} = (t', \mathsf{hk})$, $\mathsf{hk} = s_2$ とパースする. $\mathsf{msk} \leftarrow \mathsf{SS.Rec}(s_1, s_2)$, $\mathsf{dk}'_{t'} \leftarrow \mathsf{IBE.KeyGen}(\mathsf{mpk}, \mathsf{msk}, t')$ を実行し, $\mathsf{dk}_{t'} := (\mathsf{dk}'_{t'}, s_1)$ を出力する.
- $\mathsf{Enc}(\mathsf{pk}, t, \mathsf{M}) \to \mathsf{ct}_t$: $\mathsf{pk} = \mathsf{mpk}$ とパースする. $\mathsf{ct}_x \leftarrow \mathsf{IBE.Enc}(\mathsf{mpk}, t, \mathsf{M})$ を出力する.
- $\mathsf{Dec}(\mathsf{pk}, \mathsf{sk}_t, \mathsf{ct}_t) \to \mathsf{M}$ or $\perp$: $\mathsf{pk} = \mathsf{mpk}$, $\mathsf{dk}_t = (\mathsf{dk}'_t, s_1)$ とパースする. $\mathsf{M} \leftarrow \mathsf{IBE.Dec}(\mathsf{mpk}, \mathsf{dk}'_t, \mathsf{ct}_t)$ を出力する.

定理 3.3. *IBE が ℓ-IND-ID-lrCPA 安全かつ $(2, 2)$-しきい値分散法が ℓ-IND-lrAP 安全であるならば, それを用いて構成された KIE は ℓ-IND-KI-lrCPA 安全である.*

証明. 任意の攻撃者 $\mathcal{A}$ と挑戦者 $\mathcal{C}$ とのゲームを 2 つ, 以下のように定義する.

- $\mathbf{Game}_0$: 定義 3.1の IND-KI-lrCPA ゲームと同じである.
- $\mathbf{Game}_1$: 挑戦者が Setup の内部で, $(s_1, s_2) \leftarrow \mathsf{SS.Share}(|\mathsf{msk}|)$ を実行するのではなく, $(s_1, s_2) \leftarrow \mathsf{SS.Share}(0^{\mathsf{msk}})$ を実行する以外は $\mathbf{Game}_0$ と同様である.

また, $\mathbf{Game}_i$ で $\mathcal{A}$ が勝つというイベントを W_i とする.

攻撃者 $\mathcal{A}$ を用いて, $(2,2)$-しきい値分散法の ℓ-IND-lrAP 安全性を破る攻撃者 $\mathcal{D}$ を構成する.

$\mathcal{D}$ は $(\mathsf{pk}, \mathsf{dk}_0, \mathsf{hk}) \leftarrow \mathsf{Setup}(1^\lambda)$ を実行する. Setup を実行するとき内部で $(\mathsf{mpk}, \mathsf{msk}) \leftarrow \mathsf{IBE.Init}(1^\lambda)$ を実行するので msk を持っておく. $\mathcal{A}$ には pk を渡す. $(\mathsf{S}_0, \mathsf{S}_1) = (\mathsf{msk}, 0^{|\mathsf{msk}|})$ を ℓ-IND-lrAP 安全性のゲームの挑戦者 $\mathcal{C}$ に送る. 挑戦者 $\mathcal{C}$ は, $\beta \leftarrow_\$ \{0,1\}$ を取り, $(s_1, s_2) \leftarrow \mathsf{SS.Share}(\mathsf{S}_\beta)$ を実行する. $\mathcal{D}$ は $\gamma \leftarrow_\$ \{0,1\}$ を取り, その値によって $\mathcal{A}$ からのクエリの対応を変更する. 具体的には, $\gamma = 0$ であれば $\mathcal{D}$ は攻撃者 $\mathcal{A}$ が "復号鍵生成クエリと復号鍵漏洩クエリを行う攻撃者" と仮定してシミュレーションを行い, $\gamma = 1$ の場合は "復号鍵漏洩クエリを行い, その後補助鍵公開クエリを行う攻撃者" と仮定してシミュレーションを行う.

- $\gamma = 0$ の場合：各クエリに対して, $\mathcal{D}$ は以下のように振る舞う.
  - 復号鍵生成クエリ：$\mathcal{A}$ のクエリ t に対して, $\mathcal{D}$ は msk を持っているため復号鍵 dk'_t を生成可能である. また, dk'_t とペアとなる s_1 については, (まだシェア公開クエリを行っていない場合に限り) シェア公開クエリ 1 を行ってシェア s_1 を入手すれば良い.
  - 復号鍵漏洩クエリ：$\mathcal{A}$ のクエリ (t, f) に対し, 上記復号鍵生成クエリのシミュレーションと同様に $\mathsf{dk}_t = (\mathsf{dk}'_t, s_1)$ を生成し, $f(\mathsf{dk}_t)$ を返す.
  - 補助鍵公開クエリ：ゲームを中止し, $\{0,1\}$ からランダムに値を選び β' として出力する.
- $\gamma = 1$ の場合: 各クエリに対して, $\mathcal{D}$ は以下のように振る舞う.
  - 復号鍵生成クエリ：ゲームを中止し, $\{0,1\}$ からランダムに値を選び β' として出力する.
  - 復号鍵漏洩クエリ：$\mathcal{A}$ のクエリ (t, f) に対し, $\mathcal{D}$ は msk を持っているため復号鍵 dk'_t を生成可能であり, f を基に dk'_t に関する出力に対応する関数 g を構成し, $g(\mathsf{dk}'_t)$ を生成する. 次に $\mathcal{D}$ は f を基に s_1 に関する出力に対応する関数 g' を構成し, シェア漏洩クエリ $(1, g')$ を行い, $g'(s_1)$ を得る. 最終的に, $g(\mathsf{dk}'_t)$ 及び $g'(s_1)$ から $f(\mathsf{dk}_t)$ を作成し, $\mathcal{A}$ に返す.
  - 補助鍵公開クエリ：$\mathcal{D}$ は $\mathcal{C}$ に対して, シェア公開クエリ 2 を行い, 得られた s_2 を $\mathcal{A}$ に返す.

ここで, E を $\mathcal{D}$ がゲームを中止するイベントとする. E が起きる確率は, γ をランダムに選んでいるため, 次式が成り立つ.

$$\Pr[E] = \Pr[\neg E] = \frac{1}{2} \tag{1}$$

$\mathcal{A}$ がチャレンジクエリで $(t^*, \mathsf{M}_0, \mathsf{M}_1)$ を $\mathcal{D}$ にクエリすると, $\mathcal{D}$ は $b \leftarrow_\$ \{0,1\}$ を取り, $\mathsf{ct}^*_{t^*} \leftarrow \mathsf{Enc}(\mathsf{pk}, t^*, \mathsf{M}_b)$ を実行して $\mathcal{A}$ に渡す.

チャレンジクエリ終了後の復号鍵生成クエリにも同様にして対応可能である. 最後に, $\mathcal{A}$ は b の推測値として $b' \in \{0,1\}$ を出力し, $b = b'$ であるならば $\beta' = 0$, $b \neq b'$ であるならば $\beta' = 1$ として $\mathcal{D}$ は β の推測値 β' を出力する.

このようにしたときの $\mathcal{D}$ の優位性は,

$$\begin{aligned}
&\mathsf{Adv}^{\mathsf{LR\text{-}SS}}_{\Gamma,\mathcal{D},\ell,1}(\lambda) \\
&= \left|\Pr[\beta = \beta'] - \frac{1}{2}\right| \\
&= \left|\Pr[\beta = \beta' \wedge \neg E] + \Pr[\beta = \beta' \wedge E] - \frac{1}{2}\right| \\
&= \left|\Pr[\neg E]\Pr[\beta = \beta' \mid \neg E] + \Pr[E]\Pr[\beta = \beta' \mid E] - \frac{1}{2}\right| \\
&= \left|\frac{1}{2}\Pr[\beta = \beta' \mid \neg E] - \frac{1}{4}\right| \\
&= \frac{1}{4}|\Pr[b = b' \mid \beta = 0 \wedge \neg E] - \Pr[b = b' \mid \beta = 1 \wedge \neg E]| \\
&= \frac{1}{4}|\Pr[W_0] - \Pr[W_1]|.
\end{aligned}$$

ここで, 4 つ目の等号は式 (1) から成り立つ. よって,

$$|\Pr[W_0] - \Pr[W_1]| = 4\mathsf{Adv}^{\mathsf{LR\text{-}SS}}_{\Gamma,\mathcal{D},\ell,1}(\lambda) \tag{2}$$

となる.

次に, $\mathbf{Game}_1$ において, $\mathcal{A}$ を用いることで, IND-ID-lrCPA 安全な LR-IBE Π を破る攻撃者 $\mathcal{B}$ を構成する.

IND-ID-lrCPA ゲームの挑戦者 $\mathcal{C}$ は, $(\mathsf{mpk}, \mathsf{msk}) \leftarrow \mathsf{IBE.Init}(1^\lambda)$ を実行し, mpk を $\mathcal{B}$ に渡す. $\mathcal{B}$ は $\mathsf{pk} = \mathsf{mpk}$ として, $\mathcal{A}$ に pk を渡す. $\mathcal{B}$ は, $(s_1, s_2) \leftarrow \mathsf{SS.Share}(0^{|\mathsf{msk}|})$ を実行して, $\mathsf{hk} = s_2$ を持っておく. $\mathcal{B}$ は以下のようにして, $\mathcal{A}$ からのクエリに対応可能である.

- 復号鍵生成クエリ：$\mathcal{A}$ のクエリ t に対し, $\mathcal{B}$ は秘密鍵生成クエリ t を $\mathcal{C}$ に送り, 返ってきた dk'_t を用いて $\mathsf{dk}_t = (\mathsf{dk}'_t, s_1)$ を $\mathcal{A}$ に返す.
- 復号鍵漏洩クエリ：$\mathcal{A}$ のクエリ (t, f) に対し, $\mathcal{B}$ は f を基に dk'_t に関する出力に対応する関数 $g_1, \ldots, g_{\ell'}$ $(\ell' \leq \ell)$ を構成し, 秘密鍵漏洩クエリ (t, g_i) $(i \in \{1, \ldots, \ell'\})$ を $\mathcal{C}$ に行う. 得られた $g_1(\mathsf{dk}'_t), \ldots, g_{\ell'}(\mathsf{dk}'_t)$ と, f を基に s_1 に関する出力に対応する関数 g' を用いた $g'(s_1)$ から $f(\mathsf{dk}_t)$ を作成し, $\mathcal{A}$ に返す.
- 補助鍵漏洩クエリ: $\mathcal{B}$ は補助鍵 hk を持っているため, hk を $\mathcal{A}$ に返せば良い.

$\mathcal{A}$ がチャレンジクエリで $(t^*, \mathsf{M}_0, \mathsf{M}_1)$ を $\mathcal{B}$ にクエリする

と, $\mathcal{B}$ は $(t^*, \mathsf{M}_0, \mathsf{M}_1)$ を $\mathcal{C}$ にクエリする. $\mathcal{C}$ は $b \leftarrow_{\$} \{0,1\}$ を取り, $\mathsf{ct}^*_{t^*} \leftarrow \mathsf{IBE.Enc}(\mathsf{mpk}, t, \mathsf{M}_b)$ を実行し $\mathcal{B}$ に渡す. $\mathcal{B}$ は受け取った $\mathsf{ct}^*_{t^*}$ を $\mathcal{A}$ に渡す.

チャレンジクエリ終了後の復号鍵生成クエリにも同様にして対応可能である. 最後に, $\mathcal{A}$ は b の推測値として $b' \in \{0,1\}$ を出力し, $\mathcal{B}$ は受け取った b' を出力する.

このときの $\mathcal{B}$ の優位性は, 次のようになる

$$\mathsf{Adv}^{\mathsf{LR\text{-}IBE}}_{\Pi,\mathcal{B},\ell}(\lambda) = \left|\Pr[b = b'] - \frac{1}{2}\right| = \Pr[W_1] \qquad (3)$$

式 (2), (3) から,

$$\mathsf{Adv}^{\mathsf{LR\text{-}KIE}}_{\Sigma,\mathcal{A},\ell}(\lambda) \leq 4\mathsf{Adv}^{\mathsf{LR\text{-}SS}}_{\Gamma,\mathcal{D},\ell,1}(\lambda) + \mathsf{Adv}^{\mathsf{LR\text{-}IBE}}_{\Pi,\mathcal{B},\ell}(\lambda)$$

である. 従って, LR-IBE が ℓ-IND-ID-lrCPA 安全かつ $(2,2)$-しきい値分散法が ℓ-IND-lrAP 安全であるならば, それを用いて構成された KIE は ℓ-IND-KI-lrCPA 安全である. □

4. まとめと今後の課題

本稿では, 復号鍵への全体漏洩と部分漏洩, 補助鍵への全体漏洩に対して耐性のある, 漏洩耐性鍵隔離暗号の安全性を定義した. 補助鍵の部分漏洩には耐性はないものの, 現実の攻撃を考えると十分に安全な定義となっている.

今回の LR-KIE の構成は, 復号鍵漏洩クエリを一回だけ行うことができ, そのときに ℓ ビットの漏洩を許すものに対し安全性を達成するものである. しかし, より強い漏洩を行う攻撃者を考えると, 復号鍵漏洩クエリを適応的に ℓ 回行えて, 合計 ℓ ビットの漏洩を許す攻撃に対しても安全であることが理想であるので, そのような構成を見つけることが今後の課題である.

また, 復号鍵公開クエリされる場合は今回の定義で十分であるが, 補助鍵公開クエリがされる場合は, 複数の復号鍵の部分情報が漏洩しても安全であることが望ましいので, 複数の期間に ℓ ビットの漏洩を許す場合の安全性への拡張も今後の課題となる.

謝辞 本研究は JSPS 科研費 JP18H05289, JP21H03395 の助成, および文部科学省の卓越研究員事業の支援を受けたものです.

参考文献

[1] Alwen, J., Dodis, Y., Naor, M., Segev, G., Walfish, S. and Wichs, D.: Public-Key Encryption in the Bounded-Retrieval Model, *EUROCRYPT 2010*, Vol. 6110, pp. 113–134 (2010).

[2] Bellare, M. and Palacio, A.: Protecting against Key Exposure: Strongly Key-Insulated Encryption with Optimal Threshold, Cryptology ePrint Archive, Report 2002/064 (2002). https://eprint.iacr.org/2002/064.

[3] Blakley, G. R.: Safeguarding Cryptographic Keys, *Proceedings of AFIPS 1979 National Computer Conference*, Vol. 48, pp. 313–317 (1979).

[4] Boneh, D. and Franklin, M. K.: Identity-Based Encryption from the Weil Pairing, pp. 213–229.

[5] Chandran, N., Kanukurthi, B., Obbattu, S. L. B. and Sekar, S.: Adaptive Extractors and Their Application to Leakage Resilient Secret Sharing, *CRYPTO 2021, Part III*, pp. 595–624.

[6] Di Crescenzo, G., Lipton, R. J. and Walfish, S.: Perfectly Secure Password Protocols in the Bounded Retrieval Model, pp. 225–244.

[7] Dodis, Y., Katz, J., Xu, S. and Yung, M.: Key-Insulated Public Key Cryptosystems, pp. 65–82.

[8] Dziembowski, S.: Intrusion-Resilience Via the Bounded-Storage Model, pp. 207–224.

[9] Dziembowski, S. and Pietrzak, K.: Leakage-Resilient Cryptography, *49th FOCS*, pp. 293–302 (2008).

[10] Pietrzak, K.: A Leakage-Resilient Mode of Operation, *EUROCRYPT 2009* (Joux, A., ed.), Vol. 5479, pp. 462–482 (2009).

[11] Shamir, A.: Identity-Based Cryptosystems and Signature Schemes, pp. 47–53.

[12] Shamir, A.: How to Share a Secret, *Communications of the Association for Computing Machinery*, Vol. 22, No. 11, pp. 612–613 (1979).