述語暗号に対する汎用的プロキシ再暗号化方式

濱洲 青嶺[1,a)]　趙 亨駿[2]　佐藤 慎悟[2,3]　四方 順司[1,2]

概要：プロキシ再暗号化（Proxy Re-encryption，PRE）は，あるユーザの秘密鍵で復号できる暗号文を復号することなく，プロキシと呼ばれる第三者を介して，他のユーザの秘密鍵で復号できる暗号文に変換（再暗号化）する暗号技術である．Döttling と Nishimaki（PKC 2021）は，ある公開鍵暗号方式の暗号文から他の公開鍵暗号方式の暗号文に変換可能な，汎用的プロキシ再暗号化（Universal Proxy Re-encryption，UPRE）の概念を提案した．本論文では，述語暗号に対する UPRE を提案する．この UPRE の一般的構成を 2 つ示し，いずれも HRA 安全性を満たすことを証明する．1 つ目の提案方式は確率的識別不可能性難読化（probabilistic indistinguishability obfuscation, piO）から構成され，2 つ目の提案方式は Garbled Circuit から構成される．いずれの方式も対象の述語暗号方式のアルゴリズムを変更せずに再暗号化できる．

キーワード：プロキシ再暗号化，汎用的プロキシ再暗号化，述語暗号

Universal Proxy Re-encryption for Predicate Encryption

Seiryo Hamasu[1,a)]　Hyungrok Jo[2]　Shingo Sato[2,3]　Junji Shikata[1,2]

Abstract: Proxy re-encryption (PRE) is a cryptographic primitive which enables a proxy to convert a ciphertext for an user into a ciphertext for another user without knowledge of the underlying message. Döttling and Nishimaki (PKC 2021) introduced the notion of universal proxy re-encryption (UPRE). UPRE enables a proxy to convert a ciphertext of a public key encryption scheme into a ciphertext of another public key encryption scheme (possibly different from delegator's one). In this paper, we propose UPRE schemes for predicate encryption. We present two generic constructions of this UPRE and prove the HRA security of these ones. One is constructed from probabilistic indistinguishability obfuscation (piO), and the other is constructed from garbled circuit (GC). These schemes allow re-encryption without changing any algorithms of the underlying predicate encryption.

Keywords: proxy re-encryption, universal proxy re-encryption, predicate encryption

1. はじめに

1.1 背景

プロキシ再暗号化（Proxy Re-encryption，PRE) [3] は，あるユーザの秘密鍵で復号できる暗号文を，プロキシと呼ばれる第三者を介して，復号することなく他のユーザの秘密鍵で復号できる暗号文に変換（再暗号化）する．動的なアクセス権限の変更を実現する高機能暗号技術であり，電子メールの転送や分散ファイルシステムへの応用が期待されている．このため，これまで多くの PRE 方式が提案されている (e.g., [1], [2], [5], [6], [9]).

特に，Döttling と Nishimaki [8] は，PRE の新しい概念として，汎用的プロキシ再暗号化（Universal Proxy Re-encryption，UPRE）を提案した．UPRE は，任意の公開鍵暗号方式における暗号文から，必ずしも同じではない任

[1] 横浜国立大学 大学院環境情報学府/研究院
Graduate School of Environment and Information Sciences, Yokohama National University
[2] 横浜国立大学 先端科学高等研究院
Institute of Advanced Sciences, Yokohama National University
[3] 横浜国立大学 教育推進機構
Organization for the Promotion of Education, Yokohama National University
[a)] hamasu-seiryo-pw@ynu.jp

意の公開鍵暗号方式における暗号文への再暗号化を可能にする．これにより，ある公開鍵暗号方式が危殆化した場合に，その暗号文を別の公開鍵暗号方式の暗号文に変換することができる．

1.2 本論文の貢献

本論文では，[8] の UPRE の概念を拡張して，述語暗号 [11] に対する UPRE を提案する．[8] の UPRE は，任意の公開鍵暗号を再暗号化できるのに対して，提案方式は任意の述語暗号に対してプロキシによる再暗号化が可能である．よって，ID ベース暗号 [4] や属性ベース暗号 [12] のような暗号方式にも適用可能である．

- 述語暗号に対する UPRE において既存の定義はないため，この概念を導入・定式化する．特に，[8] と同様に，複数回の再暗号化が可能なマルチホップ UPRE を考える．また，この UPRE の安全性として，Honest Re-encryption Attack に対する安全性（HRA 安全性）を定式化する．

- 述語暗号に対する UPRE の一般的構成を 2 つ提案し，いずれも HRA 安全性を満たすことを証明する:
  - 1 つ目の提案方式は確率的識別不可能性難読化（Probabilistic Indistinguishability Obfuscation, piO) [7] に基づく構成である．再暗号化アルゴリズムに暗号文の属性を必要とせず，述語暗号が属性を秘匿する場合，UPRE に用いてもその性質は保持される．また，再暗号化の回数に制限のないマルチホップ UPRE である．
  - 2 つ目の方式は，Garbled Circuit（GC）からの構成である．piO に基づく構成と同様に，属性の秘匿性を保持する．この方式は, 1 つ目の提案方式よりも弱い暗号プリミティブを用いるマルチホップ UPRE であるが，再暗号化の度に暗号文サイズと復号の計算量が増大するため，再暗号化の回数には実質的な制限がある．

これらの提案方式は, 対象の述語暗号方式のアルゴリズムを変更することなく, プロキシによる再暗号化を実行することができる. つまり, 各ユーザに負担の大きい処理を付加することなく, 強力な計算能力をもつプロキシに再暗号化の処理を委託することができる.

1.3 論文の構成

第 2 節では，本論文で利用する表記法と前提となる暗号技術について述べる．第 3 節では，述語暗号に対する UPRE とその安全性に関して定式化する．第 4 節，第 5 節では，それぞれ piO，GC に基づく構成を提案し，安全性証明を与える．最後に第 6 節で本論文を総括する．

2. 準備

2.1 記法

本論文では, 確率的多項式時間アルゴリズムを PPT アルゴリズムと呼ぶ．また，$n \in \mathbb{N}$ に対して $[n] := \{1, \cdots, n\}$ と定義する．n 個の値 $x_1, \ldots, x_n$ に対して $(n \in \mathbb{N})$, $\{x_i\}_{i=1}^{n} := \{x_1, \ldots, x_n\}$ と定義する．λ に関する多項式を $\mathsf{poly}(\lambda)$ と表記する．さらに，関数 $f : \mathbb{N} \to \mathbb{R}$ が任意の定数 $c > 0$ と十分に大きな $\lambda \in \mathbb{N}$ に対して $f(\lambda) < \lambda^{-c}$ であるとき，f は無視可能関数であるといい，$\mathsf{negl}(\lambda)$ で表す．集合 X から要素を一様ランダムに選ぶことを $x \leftarrow\!\!\$\ X$, アルゴリズム A が入力 x に対して y を出力することを $y \leftarrow A(x)$ と表記する．

2.2 確率的識別不可能難読化

定義 1 (確率的識別難読化)**.** [7] 関数 C を入力にとり，C を難読化した関数 $\bar{C}$ を出力する PPT アルゴリズム piO が以下のような性質を持つとき，確率的識別不可能難読化（piO）である．

正当性. 任意の関数 C に対して, $\bar{C} \leftarrow \mathsf{piO}(C)$ とする．このとき，入力 x に対して，$C(x)$ と $\bar{C}(x)$ が従う分布は等しい．

識別不可能性. ある分布に従ってサンプリングされた 2 つの関数 C_0, C_1 に対して，任意の PPT アルゴリズム $\mathcal{D}$ が $\Pr[\mathcal{D}(\mathsf{piO}(C_1)) = \mathcal{D}(\mathsf{piO}(C_2))] \geq 1 - \mathsf{negl}(\lambda)$ を満たす．

2.3 Garbled Circuit

定義 2 (Garbled Circuit)**.** Garbled Circuit（GC）方式は, 2 つの PPT アルゴリズム $(\mathsf{Garble}, \mathsf{Eval})$ から成る．

- $(\bar{C}, \{\mathsf{label}_{i,b}\}_{i\in[n],b\in\{0,1\}}) \leftarrow \mathsf{Garble}(1^\lambda, C)$: セキュリティパラメータ 1^λ と n ビットを入力にとる回路 C を入力にとり，ガーブルド回路 $\bar{C}$ と $2n$ 個のラベル $\{\mathsf{label}_{i,b}\}_{i\in[n],b\in\{0,1\}}$ を出力する．

- $y \leftarrow \mathsf{Eval}(\bar{C}, \{\mathsf{label}_i\}_{i\in[n]})$: ガーブルド回路 $\bar{C}$ と n 個のラベル $\mathsf{label} := \{\mathsf{label}_i\}_{i\in[n]}$ を入力にとり，回路の計算結果 y を出力する．

正当性. 入力が n である任意の回路 C に対して，$(\bar{C}, \{\mathsf{label}_{i,b}\}_{i\in[n],b\in\{0,1\}}) \leftarrow \mathsf{Garble}(1^\lambda, C)$ とする．このとき, 任意の入力 $x := (x_1, \cdots, x_n) \in \{0,1\}^n$ に対して，$\Pr[\mathsf{Eval}(\bar{C}, \{\mathsf{label}_{i,x_i}\}_{i\in[n]}) = C(x)] \geq 1 - \mathsf{negl}(\lambda)$ が成り立つとき，GC 方式は正当である．

安全性. Sim を PPT アルゴリズムとする．チャレンジャー $\mathcal{C}$ と攻撃者 $\mathcal{A}$ の間で行われる次のゲームを考える．

(1) $\mathcal{C}$ はランダムビット $b \leftarrow\!\!\$\ \{0,1\}$ を選び，セキュリティパラメータ 1^λ を $\mathcal{A}$ に与える．

(2) $\mathcal{A}$ は $\mathcal{C}$ に回路 C とその入力 $x \in \{0,1\}^n$ を送信する．$b=0$ のとき，$\mathcal{C}$ は $(\bar{C}, \{\mathsf{label}_{i,b}\}_{i\in[n],b\in\{0,1\}}) \leftarrow \mathsf{Garble}(1^\lambda, C)$ を計算し，$\bar{C}$ と $\{\mathsf{label}_{i,x_i}\}_{i\in[n]}$ を $\mathcal{A}$ に返す．$b=1$ のとき，$(\bar{C}, \{\mathsf{label}_i\}_{i\in[n]}) \leftarrow \mathsf{Sim}(1^\lambda, C(x))$ を $\mathcal{A}$ に返す．

(3) $\mathcal{A}$ は b の値を推測し，$b' \in \{0,1\}$ を出力する．

このゲームにおいて，任意の PPT アルゴリズム $\mathcal{A}$ に対して $|\Pr[b=b'] - 1/2| \leq \mathsf{negl}(\lambda)$ であるような Sim が存在するならば，GC は選択的モデルにおいて安全である．

2.4 秘密分散

定義 3 (秘密分散)．平文空間 $\mathcal{M}$ に対する (t,n)-秘密分散方式は，以下に示す 2 つの PPT アルゴリズム $(\mathsf{Share}, \mathsf{ReCon})$ から成る．

- $(s_1, \cdots, s_n) \leftarrow \mathsf{Share}(1^\lambda, m)$：セキュリティパラメータ 1^λ と平文 $m \in \mathcal{M}$ を入力にとり，n 個のシェア $(s_1, \cdots, s_n)$ を出力する．
- $m \leftarrow \mathsf{ReCon}(s_{i_1}, \cdots, s_{i_t})$：$t$ 個のシェア $(s_{i_1}, \cdots, s_{i_t})$ を入力にとり，平文 m' を出力する．

正当性． 任意の $m \in \mathcal{M}$，要素数 t の集合 $\{i_1, \cdots, i_t\} \subseteq [n]$ 及び $(s_1, \cdots, s_n) \leftarrow \mathsf{Share}(1^\lambda, m)$ に対して，$\Pr[\mathsf{ReCon}(s_{i_1}, \cdots, s_{i_t}) = m] \geq 1 - \mathsf{negl}(\lambda)$ が成り立つ．

安全性． 任意の $m, m' \in \mathcal{M}$ に対して，$(s_1, \cdots, s_n) \leftarrow \mathsf{Share}(1^\lambda, m)$，$(s'_1, \cdots, s'_n) \leftarrow \mathsf{Share}(1^\lambda, m')$ とする．このとき，$|S| < t$ を満たす任意の $S \subseteq [n]$ に対して，$\{s_i\}_{i\in[S]}$ と $\{s'_i\}_{i\in[S]}$ は統計的に識別不可能である．

2.5 （弱い）バッチ暗号化

定義 4 (バッチ暗号化 [8])．平文空間 $\mathcal{M}$ に対する（弱い）バッチ暗号化方式は，以下に示す 3 つの PPT アルゴリズム $(\mathsf{BGen}, \mathsf{BEnc}, \mathsf{BDec})$ から成る．

- $(\hat{\mathsf{pk}}, \hat{\mathsf{sk}}) \leftarrow (1^\lambda, s)$：セキュリティパラメータ 1^λ とビット列 $b \in \{0,1\}^\lambda$ を入力にとり，公開鍵 $\hat{\mathsf{pk}}$ と秘密鍵 $\hat{\mathsf{sk}}$ を出力する．
- $\hat{\mathsf{ct}} \leftarrow \mathsf{BEnc}(\hat{\mathsf{pk}}, \{(m_{i.0}, m_{i.1})\}_{i\in[\lambda]})$：公開鍵 $\hat{\mathsf{pk}}$ と $\mathcal{M}$ の要素である λ 組の平文のペア $\{(m_{0.i}, m_{1,i})\}_{i\in[\lambda]}$ を入力にとり，暗号文 $\hat{\mathsf{ct}}$ を出力する．
- $\{m_i\}_{i\in[\lambda]} \leftarrow \mathsf{BDec}(\hat{\mathsf{sk}}, \hat{\mathsf{ct}})$：秘密鍵 $\hat{\mathsf{sk}}$ と暗号文 $\hat{\mathsf{ct}}$ を入力にとり，λ 個の平文 $\{m_i\}_{i\in[\lambda]}$ を出力する．

正当性． 任意の $\lambda \in \mathbb{N}$，$s := (s_1, \cdots, s_\lambda) \in \{0,1\}^\lambda$ 及び平文 $m_{i.b} \in \mathcal{M}$ に対して，$(\hat{\mathsf{pk}}, \hat{\mathsf{sk}}) \leftarrow (1^\lambda, s)$，$\hat{\mathsf{ct}} \leftarrow \mathsf{BEnc}(\hat{\mathsf{pk}}, \{(m_{0.i}, m_{1,i})\}_{i\in[\lambda]})$ とする．このとき，$\Pr[\{m_{i.s_i}\}_{i\in[\lambda]} = \mathsf{BDec}(\hat{\mathsf{sk}}, \hat{\mathsf{ct}})] \geq 1 - \mathsf{negl}(\lambda)$ が成り立つ．

安全性． チャレンジャー $\mathcal{C}$ と攻撃者 $\mathcal{A}$ の間で行われる以下のゲームを考える．

(1) $\mathcal{A}$ は $s := (s_1, \cdots, s_\lambda) \in \{0,1\}^\lambda$ を選び，$\mathcal{C}$ に与える．

(2) $\mathcal{C}$ はランダムビット $b \leftarrow\!\!\$\ \{0,1\}$ を選ぶ．また，$(\hat{\mathsf{pk}}, \hat{\mathsf{sk}}) \leftarrow \mathsf{BGen}(1^\lambda, s)$ を計算し，$\hat{\mathsf{pk}}$ を $\mathcal{A}$ に与える．

(3) $\mathcal{A}$ は，$\mathcal{C}$ に平文 $\{(m_{i.0}, m_{i.1})\}_{i\in[\lambda]}$ を送信する．$\mathcal{C}$ は，$b=0$ ならば $\hat{\mathsf{ct}}^* \leftarrow \mathsf{BEnc}(\hat{\mathsf{pk}}, \{(m_{i.0}, m_{i.1})\}_{i\in[\lambda]})$，$b=1$ ならば $\hat{\mathsf{ct}}^* \leftarrow \mathsf{BEnc}(\hat{\mathsf{pk}}, \{(m_{i.s_i}, m_{i.s_i})\}_{i\in[\lambda]})$ を計算する．そして，$\mathcal{A}$ に $\hat{\mathsf{ct}}^*$ を返す．

(4) $\mathcal{A}$ は b の値を推測し，$b' \in \{0,1\}$ を出力する．

上記のゲームにおいて，任意の PPT アルゴリズム $\mathcal{A}$ に対して，$|\Pr[b=b'] - 1/2| \leq \mathsf{negl}(\lambda)$ が成り立つ．

上記の弱いバッチ暗号化方式は，IND-CPA 安全な公開鍵暗号方式から構成できることが知られている [8]．

2.6 述語暗号

述語暗号（Predicate Encryption，PE）のシンタックスと，その CPA 安全性を定義する．

定義 5 (述語暗号)．[10] 集合 $\mathcal{X}, \mathcal{Y}$ に対して，述語 $\mathsf{P} : \mathcal{X} \times \mathcal{Y} \to \{0,1\}$ を定める．また，平文空間を $\mathcal{M}$ とする．このとき，P に対する述語暗号（PE）方式は，以下に示す 4 つのアルゴリズムの組 $(\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ で定義される．

- $(\mathsf{pp}, \mathsf{msk}) \leftarrow \mathsf{Setup}(1^\lambda)$：セキュリティパラメータ 1^λ を入力にとり，公開パラメータ pp 及びマスター秘密鍵 msk を出力する．
- $\mathsf{sk}_x \leftarrow \mathsf{KeyGen}(\mathsf{pp}, \mathsf{msk}, x)$：公開パラメータ pp，マスター秘密鍵 msk 及び $x \in \mathcal{X}$ を入力にとり，秘密鍵 sk_x を出力する．
- $\mathsf{ct}_y \leftarrow \mathsf{Enc}(\mathsf{pp}, y, m)$：公開パラメータ pp，$y \in \mathcal{Y}$ 及び平文 $m \in \mathcal{M}$ を入力にとり，暗号文 ct_y を出力する．
- $m \,/\, \perp \leftarrow \mathsf{Dec}(\mathsf{pp}, \mathsf{sk}_x, \mathsf{ct}_y)$：公開パラメータ pp，秘密鍵 sk_x 及び暗号文 ct_y を入力にとり，平文 $m \in \mathcal{M}$ またはエラーシンボル $\perp$ を出力する．

正当性． 任意の $\lambda \in \mathbb{N}$，$(\mathsf{pp}, \mathsf{msk}) \leftarrow \mathsf{Setup}(1^\lambda)$，平文 $m \in \mathcal{M}$ 及び $\mathsf{P}(x,y) = 1$ を満たす $x \in \mathcal{X}$，$y \in \mathcal{Y}$ に対して，$\mathsf{sk}_x \leftarrow \mathsf{KeyGen}(\mathsf{pp}, \mathsf{msk}, x)$，$\mathsf{ct}_y \leftarrow \mathsf{Enc}(\mathsf{pp}, y, m)$ とする．このとき，$\Pr[\mathsf{Dec}(\mathsf{pp}, \mathsf{sk}_x, \mathsf{ct}_y) = m] \geq 1 - \mathsf{negl}(\lambda)$ が成り立つ．

安全性． PE の安全性は，チャレンジャー $\mathcal{C}$ と攻撃者 $\mathcal{A}$ の間で行われる以下のゲームによって定義される．

Init. $\mathcal{A}$ は $y^* \in \mathcal{Y}$ を $\mathcal{C}$ に与える．

Setup. $\mathcal{C}$ は $(\mathsf{pp}, \mathsf{msk}) \leftarrow \mathsf{Setup}(1^\lambda)$ を実行し，$\mathcal{A}$ に

pp を与える．

Query Phase 1. $\mathcal{A}$ は鍵生成オラクルに任意の回数アクセスできる．鍵生成オラクルは，$x \in \mathcal{X}$ を入力にとり，$\mathsf{P}(x, y^*) = 1$ ならば $\perp$ を出力する．それ以外の場合，$\mathsf{sk}_x \leftarrow \mathsf{KeyGen}(\mathsf{pp}, \mathsf{msk}, x)$ を出力する．

Challenge. $\mathcal{A}$ は 2 つの平文 $(m_0, m_1) \in \mathcal{M} \times \mathcal{M}$ を $\mathcal{C}$ に与える．$\mathcal{C}$ はランダムビット $b \leftarrow\!\!\$\ \{0,1\}$ を選び，$\mathsf{ct}^* \leftarrow \mathsf{Enc}(\mathsf{pp}, y^*, m_b)$ を $\mathcal{A}$ に与える．

Query Phase 2. Query Phase 1 と等しい．

Guess. $\mathcal{A}$ は b の値を推測し，$b' \in \{0,1\}$ を出力する．

上のゲームにおいて，任意の PPT アルゴリズム $\mathcal{A}$ に対して $\mathsf{Adv}^{\mathcal{A}}_{\mathsf{PE}}(\lambda) := |\Pr[b' = b] - 1/2| \leq \mathsf{negl}(\lambda)$ が成り立つならば，PE 方式は選択的モデルにおいて CPA 安全である．

以降，平文空間 $\mathcal{M}_i$ と述語 $\mathsf{P}_i : \mathcal{X}_i \times \mathcal{Y}_i \to \{0,1\}$ に対する PE 方式を，$\Sigma^{\mathsf{PE}}_i = (\mathsf{Setup}_i, \mathsf{KeyGen}_i, \mathsf{Enc}_i, \mathsf{Dec}_i)$ と表記する．

3. 汎用的述語プロキシ再暗号化

本節では，UPRE の概念 [8] を PE に拡張した汎用的述語プロキシ再暗号化（Universal Predicate Proxy Re-encryption，UPPRE）について定式化する．

3.1 UPPRE のモデル

定義 6 (UPPRE)．PE 方式 $\Sigma^{\mathsf{PE}}_i, \Sigma^{\mathsf{PE}}_j$ に対して，UPPRE 方式は，以下に示す 3 つの PPT アルゴリズム $(\mathsf{RKG}, \mathsf{REnc}, \mathsf{RDec})$ から成る．

- $\mathsf{rk}_{i.x \to j.x'} \leftarrow \mathsf{RKG}(1^\lambda, \Sigma^{\mathsf{PE}}_i, \Sigma^{\mathsf{PE}}_j, x, x', \mathsf{pp}_i, \mathsf{sk}_{i.x}, \mathsf{pp}_j)$: セキュリティパラメータ 1^λ，PE 方式 $\Sigma^{\mathsf{PE}}_i, \Sigma^{\mathsf{PE}}_j$，$(x, x') \in \mathcal{X}_i \times \mathcal{X}_j$，$\Sigma^{\mathsf{PE}}_i$ の公開パラメータ pp_i と秘密鍵 $\mathsf{sk}_{i.x}$ 及び Σ^{PE}_j の公開パラメータ pp_j を入力にとり，再暗号鍵 $\mathsf{rk}_{i.x \to j.x'}$ を出力する．
- $\mathsf{ct}_{j.y'} \leftarrow \mathsf{REnc}(\Sigma^{\mathsf{PE}}_i, \Sigma^{\mathsf{PE}}_j, \mathsf{rk}_{i.x \to j.x'}, \mathsf{ct}_{i.y})$: $\Sigma^{\mathsf{PE}}_i, \Sigma^{\mathsf{PE}}_j$，再暗号鍵 $\mathsf{rk}_{i.x \to j.x'}$ 及び Σ_i における $y \in \mathcal{Y}_i$ の暗号文 $\mathsf{ct}_{i.y}$ を入力にとり，Σ_j における $y' \in \mathcal{Y}_j$ の暗号文 $\mathsf{ct}_{j.y'}$ を出力する．
- $m \leftarrow \mathsf{RDec}(\Sigma^{\mathsf{PE}}_j, \mathsf{pp}_j, \mathsf{sk}_{j.x'}, \mathsf{ct}_{j.y'})$: Σ^{PE}_j，pp_j，$\mathsf{sk}_{j.x'}$ 及び再暗号化された暗号文 $\mathsf{ct}_{j.y'}$ を入力にとり，平文 $m \in \mathcal{M}_j$ を出力する．

定義 7 (正当性)．$L \in \mathbb{N}$ に対して，任意の PE 方式 $\Sigma^{\mathsf{PE}}_0, \Sigma^{\mathsf{PE}}_1, \cdots, \Sigma^{\mathsf{PE}}_L$ を考える．まず，任意の $\mathsf{P}_0(x_0, y_0) = 1$ を満たす $x_0 \in \mathcal{X}_0, y_0 \in \mathcal{Y}_0$ 及び $m \in \mathcal{M}_0 \cap \cdots \cap \mathcal{M}_L$ に対して，$(\mathsf{pp}_0, \mathsf{msk}_0) \leftarrow \mathsf{Setup}_0(1^{\lambda_0})$，$\mathsf{sk}_{0.x_0} \leftarrow \mathsf{KeyGen}_0(\mathsf{pp}_0, \mathsf{msk}_0, x_0)$，$\mathsf{ct}_{0.y_0} \leftarrow \mathsf{Enc}_0(\mathsf{pp}_0, y_0, m)$ とする．次に，全ての $l \in [L]$ と任意の $x_l \in \mathcal{X}_l$ について，

$(\mathsf{pp}_l, \mathsf{msk}_l) \leftarrow \mathsf{Setup}_l(1^{\lambda_l})$，

$\mathsf{sk}_{l.x_l} \leftarrow \mathsf{KeyGen}_l(\mathsf{pp}_l, \mathsf{msk}_l, x_l)$，　$\mathsf{rk}_{l-1.x_{l-1} \to l.x_l} \leftarrow \mathsf{RKG}(1^\lambda, \Sigma^{\mathsf{PE}}_{l-1}, \Sigma^{\mathsf{PE}}_l, x_{l-1}, x_l, \mathsf{pp}_{l-1}, \mathsf{sk}_{l-1.x_{l-1}}, \mathsf{pp}_l)$，

$\mathsf{ct}_{l.y_l} \leftarrow \mathsf{REnc}(\Sigma^{\mathsf{PE}}_{l-1}, \Sigma^{\mathsf{PE}}_l, \mathsf{rk}_{l-1.x_{l-1} \to l.x_l}, \mathsf{ct}_{l-1.y_{l-1}})$

を計算する．このとき，全ての $l \in [L]$ に対して

$$\Pr[\mathsf{RDec}(\Sigma^{\mathsf{PE}}_l, \mathsf{pp}_l, \mathsf{sk}_{l.x_l}, \mathsf{ct}_{l.y_l}) = m] \geq 1 - \mathsf{negl}(\lambda)$$

が成り立つならば，UPPRE 方式は正当である．特に，$L > 1$ であるとき，UPPRE 方式はマルチホップであるという．

3.2 安全性定義

本論文では，UPPRE の安全性要件として，HRA 安全性 [5] を定義する．HRA 安全性は PRE における CPA 安全性を強化した安全性であり，CPA 安全性を含意する．HRA 安全性に対する攻撃者は，CPA 安全性ゲームと同様に再暗号化鍵オラクルが与えられる．それに加えて，暗号化オラクルで生成された暗号文に対して，再暗号化オラクルを用いて変換された暗号文を入手することができる．

定義 8 (HRA 安全性)．UPPRE の安全性は，チャレンジャー $\mathcal{C}$ と攻撃者 $\mathcal{A}$ の間で行われる以下のゲームによって定義される．

Init. $N := \mathsf{poly}(\lambda)$ に対して，$\mathcal{A}$ は N 個の PE 方式 $\{\Sigma^{\mathsf{PE}}_i\}_{i=1}^N$ とターゲットの組 $(y^*_1, \cdots, y^*_N) \in \mathcal{Y}_1 \times \cdots \times \mathcal{Y}_N$ を選ぶ．そして，$\mathcal{C}$ に $\{\Sigma^{\mathsf{PE}}_i, y^*_i\}_{i=1}^N$ を与える．

Setup. 各 $i \in [N]$ に対して，$\mathcal{C}$ は $(\mathsf{pp}_i, \mathsf{msk}_i) \leftarrow \mathsf{Setup}_i(1^\lambda)$ を実行し，$\mathcal{A}$ に $\{\mathsf{pp}_i\}_{i=1}^N$ を与える．また，集合 $H_{\mathsf{ct}}, H_{\mathsf{derive}} \leftarrow \phi$ 及びカウンタ $\#\mathsf{CT} \leftarrow 0$ を定義する．

Query Phase 1. $\mathcal{A}$ は順番と回数を問わず，以下のオラクルにアクセスできる．

- 暗号化オラクル $\mathcal{O}^{\mathsf{Enc}}(i, y, m) : i \in [N]$，$y \in \mathcal{Y}_i$ 及び平文 $m \in \mathcal{M}_1 \cap \cdots \cap \mathcal{M}_N$ を入力にとり，$\mathsf{ct}_{i.y} \leftarrow \mathsf{Enc}(\mathsf{pp}_i, y, m)$ を出力する．また，$\#\mathsf{CT} \leftarrow \#\mathsf{CT} + 1$ とインクリメントし，$H_{\mathsf{ct}} \leftarrow H_{\mathsf{ct}} \cap \{(\#\mathsf{CT}, i, y, \mathsf{ct}_{i.y})\}$ とする．
- 鍵生成オラクル $\mathcal{O}^{\mathsf{KeyGen}}(i, x) : i \in [N]$ 及び $x \in \mathcal{X}_i$ を入力にとり，$\mathsf{P}_i(x, y^*_i) = 1$ ならば $\perp$ を出力する．それ以外の場合，$\mathsf{sk}_{i.x} \leftarrow \mathsf{KeyGen}_i(\mathsf{pp}_i, \mathsf{msk}_i, x)$ を出力する．
- 再暗号化鍵オラクル $\mathcal{O}^{\mathsf{RKG}}(i, j, x, x') :$ $(i, j) \in [N] \times [N]$ 及び $(x, x') \in \mathcal{X}_i \times \mathcal{X}_j$ を入力にとり，$i = j$ または $\mathsf{P}_i(x, y^*_i) = 1 \wedge \mathsf{P}_j(x', y^*_j) \neq 1$ ならば $\perp$ を出力する．それ以外の場合，$\mathsf{sk}_{i.x} \leftarrow \mathsf{KeyGen}_i(\mathsf{pp}_i, \mathsf{msk}_i, x)$ を計算し，$\mathsf{rk}_{i.x \to j.x'} \leftarrow \mathsf{RKG}(\Sigma^{\mathsf{PE}}_i, \Sigma^{\mathsf{PE}}_j, x, x', \mathsf{sk}_{i.x}, \mathsf{pp}_j)$ を出力する．
- 再暗号化オラクル $\mathcal{O}^{\mathsf{REnc}}(i, j, x, x', k) :$ $(i, j) \in$

$[N] \times [N]$ 及び $k \leq \#\mathsf{CT}$ を入力にとり，$i = j$ ならば $\perp$ を出力する．また，ある $y \in \mathcal{Y}_i$ 及び $\mathsf{ct}_{i.y}$ に対して，$(k, i, y, \mathsf{ct}_{i.y}) \notin H_{\mathsf{ct}}$ ならば $\perp$ を出力する．それ以外の場合，$\mathsf{sk}_{i.x} \leftarrow \mathsf{KeyGen}_i(\mathsf{pp}_i, \mathsf{msk}_i, x)$ 及び $\mathsf{rk}_{i.x \to j.x'} \leftarrow \mathsf{RKG}(\Sigma_i^{\mathsf{PE}}, \Sigma_j^{\mathsf{PE}}, x, x', \mathsf{sk}_{i.x}, \mathsf{pp}_j)$ を計算し，$\mathsf{ct}_{j.y'} \leftarrow \mathsf{REnc}(\Sigma_i^{\mathsf{PE}}, \Sigma_j^{\mathsf{PE}}, \mathsf{rk}_{i.x \to j.x'}, \mathsf{ct}_{i.y})$ を出力する．そして，$\#\mathsf{CT} \leftarrow \#\mathsf{CT} + 1$ とインクリメントし，$H_{\mathsf{ct}} \leftarrow H_{\mathsf{ct}} \cap \{(\#\mathsf{CT}, j, y', \mathsf{ct}_{j.y'})\}$ とする．

Challenge. $\mathcal{A}$ は $i^* \in [N]$ 及び平文 $(m_0, m_1) \in \mathcal{M}_1 \cap \cdots \cap \mathcal{M}_N$ を $\mathcal{C}$ に与える．$\mathcal{C}$ はランダムに $b \leftarrow\!\!\$\ \{0,1\}$ を選び，$\mathcal{A}$ に $\mathsf{ct}^* \leftarrow \mathsf{Enc}_{i^*}(\mathsf{pp}_{i^*}, y^*_{i^*}, m_b)$ を与える．また，$\#\mathsf{CT} \leftarrow \#\mathsf{CT} + 1$ とインクリメントし，$H_{\mathsf{ct}} \leftarrow H_{\mathsf{ct}} \cap \{(\#\mathsf{CT}, i^*, y^*, \mathsf{ct}^*)\}$ とする．さらに，$H_{\mathsf{derive}} \leftarrow H_{\mathsf{derive}} \cap \{\#\mathsf{CT}\}$ とする．

Query Phase 2. Query Phase 1 と同様に，$\mathcal{A}$ はオラクルにアクセスできる．ただし，$\mathcal{O}^{\mathsf{REnc}}(i, j, x, x', k)$ において，以下の処理を追加する．

- $k \in H_{\mathsf{derive}}$ かつ $\mathsf{P}_j(x', y^*_j) \neq 1$ ならば $\perp$ を出力する．
- $k \in H_{\mathsf{derive}}$ かつ出力が $\perp$ でない場合，$H_{\mathsf{derive}} \leftarrow H_{\mathsf{derive}} \cap \{\#\mathsf{CT}\}$ とする．

Guess. $\mathcal{A}$ は b の値を推測し，$b' \in \{0,1\}$ を出力する．

上記のゲームにおいて，任意の $N = \mathsf{poly}(\lambda)$ 及び PPT アルゴリズム $\mathcal{A}$ に対して $\mathsf{Adv}^{\mathcal{A}}_{\mathsf{UPPRE}}(\lambda) := |\Pr[b = b'] - 1/2| \leq \mathsf{negl}(\lambda)$ が成り立つならば，UPPRE 方式は選択的モデルにおいて HRA 安全である．

4. piO を用いた UPPRE 方式

本節では，piO に基づく UPPRE 方式を提案する．再暗号化鍵 $\mathsf{rk}_{i.x \to j.x'}$ は，図 1 の再暗号化関数 $\mathsf{C}^{\mathsf{REnc}}_{i.x \to j.x'}$ を難読化した関数となっている．再暗号化アルゴリズムでは，この関数に暗号文 $\mathsf{ct}_{i.y}$ を与え，内部で $\mathsf{sk}_{i.x}$ を用いて復号し，pp_j を用いて再度暗号化することで，再暗号化を実現する．

4.1 提案方式の構成

piO を piO 方式，$\Sigma_i^{\mathsf{PE}}, \Sigma_j^{\mathsf{PE}}$ を PE 方式であるとする．このとき，UPPRE 方式 $\mathsf{UPPRE}_1 = (\mathsf{RKG}, \mathsf{REnc}, \mathsf{RDec})$ の構成は以下に示す通りである．

- $\mathsf{rk}_{i.x \to j.x'} \leftarrow \mathsf{RKG}(1^\lambda, \Sigma_i^{\mathsf{PE}}, \Sigma_j^{\mathsf{PE}}, x, x', \mathsf{pp}_i, \mathsf{sk}_{i.x}, \mathsf{pp}_j)$:
  (1) $\mathsf{P}_j(x', y') = 1$ を満たす $y' \leftarrow\!\!\$\ \mathcal{Y}_j$ を選ぶ．
  (2) 図 1 に示す関数 $\mathsf{C}^{\mathsf{REnc}}_{i.x \to j.x'}$ を構成する．
  (3) $\mathsf{rk}_{i.x \to j.x'} := \bar{\mathsf{C}}^{\mathsf{REnc}}_{i.x \to j.x'} \leftarrow \mathsf{piO}(\mathsf{C}^{\mathsf{REnc}}_{i.x \to j.x'})$ を出力する．

- $\mathsf{ct}_{j.y'} \leftarrow \mathsf{REnc}(\Sigma_i^{\mathsf{PE}}, \Sigma_j^{\mathsf{PE}}, \mathsf{rk}_{i.f \to j.g}, \mathsf{ct}_{i.y})$:
  (1) $\mathsf{rk}_{i.x \to j.x'} := \bar{\mathsf{C}}^{\mathsf{REnc}}_{i.x \to j.x'}$ とする．
  (2) $\mathsf{ct}_{j.y'} \leftarrow \bar{\mathsf{C}}^{\mathsf{REnc}}_{i.x \to j.x'}(\mathsf{ct}_{i.y})$ を出力する．

再暗号化関数 $\mathsf{C}^{\mathsf{REnc}}_{i.x \to j.x'}$:
入力：$\mathsf{ct}_{i.y}$
定数：$\mathsf{sk}_{i.x}$, pp_i, pp_j, y'
(1) $m' \leftarrow \mathsf{Dec}_i(\mathsf{pp}_i, \mathsf{sk}_{i.x}, \mathsf{ct}_{i.y})$ を計算する．
(2) $\mathsf{ct}_{j.y'} \leftarrow \mathsf{Enc}_j(\mathsf{pp}_j, y', m')$ を出力する．

図 1 再暗号化関数 $\mathsf{C}^{\mathsf{REnc}}_{i.x \to j.x'}$
Fig. 1 Re-encryption Function $\mathsf{C}^{\mathsf{REnc}}_{i.x \to j.x'}$

- $m \leftarrow \mathsf{RDec}(\mathsf{pp}_j, \mathsf{sk}_{j.x'}, \mathsf{ct}_{j.y'})$:
  (1) $m \leftarrow \mathsf{Dec}_j(\mathsf{pp}_j, \mathsf{sk}_{j.x'}, \mathsf{ct}_{j.y'})$ を出力する．

4.2 正当性

任意の暗号文 $\mathsf{ct}_{i.y} \leftarrow \mathsf{Enc}_i(\mathsf{pp}_i, y, m)$ と $\mathsf{rk}_{i.x \to j.x'} := \mathsf{piO}(\mathsf{C}^{\mathsf{REnc}}_{i.x \to j.x'})$ に対して，再暗号化を計算する．$\mathsf{P}_i(x, y) = 1$ のとき，$\mathsf{C}^{\mathsf{REnc}}_{i.x \to j.x'}(\mathsf{ct}_{i.y})$ の出力は $\mathsf{Enc}_j(\mathsf{pp}_j, y', m)$ と同分布なので，piO 方式の正当性より，$\mathsf{ct}_{j.y'} \leftarrow \mathsf{REnc}(\Sigma_i^{\mathsf{PE}}, \Sigma_j^{\mathsf{PE}}, \mathsf{rk}_{i.f \to j.g}, \mathsf{ct}_{i.y})$ も $\mathsf{Enc}_j(\mathsf{pp}_j, y', m)$ と同分布である．$\mathsf{P}_j(x', y') = 1$ より，$\mathsf{RDec}(\mathsf{pp}_j, \mathsf{sk}_{j.x'}, \mathsf{ct}_{j.y'}) = \mathsf{Dec}_j(\mathsf{pp}_j, \mathsf{sk}_{j.x'}, \mathsf{ct}_{j.y'}) = m$ であるから，復号結果は正当である．さらに，$\mathsf{ct}_{j.y'}$ は Σ_j^{PE} の暗号文空間に含まれるので，$\mathsf{ct}_{j.y'}$ に対しても同様に再暗号化を行うことができる．したがって，UPPRE_1 は任意回数の再暗号化に対応するマルチホップ UPPRE である．

4.3 安全性証明

定理 1. $N := \mathsf{poly}(\lambda)$ とする．このとき，各 $i \in [N]$ に対して，PE 方式 Σ_i^{PE} が CPA 安全で，かつ piO が piO の正当性と安全性を満たすならば，UPPRE 方式 UPPRE_1 は選択的モデルにおいて HRA 安全である．

証明． UPPRE_1 の HRA 安全性を証明するために，以下のゲーム列を考える．ここで，$\mathcal{A}$ を提案方式 UPPRE_1 に対する PPT 攻撃者とし，Game t において $\mathcal{A}$ が $b' = b$ となる $b' \in \{0,1\}$ を出力する確率を Win_t と表記する．

Game 0. 定義 8 に示す通常の安全性ゲームである．このとき，$\mathsf{Adv}^{\mathcal{A}}_{\mathsf{UPPRE}_1}(\lambda) = |\mathsf{Win}_0 - 1/2|$ である．

Game 1. Query Phase の $\mathcal{O}^{\mathsf{REnc}}$ や Challenge において，$\mathsf{ct}_{i.y} \leftarrow \mathsf{Enc}_i(\mathsf{pp}_i, y, m)$ を実行するとき，H_{ct} に格納する値に平文 m を追加する．それ以外は Game 0 と同一である．$\mathcal{A}$ から見て Game 0 と Game 1 は同一なので，$\mathsf{Win}_1 = \mathsf{Win}_0$ である．

Game 2. 再暗号化オラクル $\mathcal{O}^{\mathsf{REnc}}(i, j, x, x', k)$ において，出力が $\perp$ でないときの再暗号化の計算方法を変更する．$\mathsf{P}_i(x, y^*_i) = 1$ ならば，REnc を実行する代わりに，H_{ct} の中から $(k, i, y, \mathsf{ct}_{i.y}, m)$ を探し，m を取り出す．そして，$\mathsf{P}_j(x', y') = 1$ を満たす $y \in \mathcal{Y}_j$ をランダムに選び，$\mathsf{ct}_{j.y'} \leftarrow \mathsf{Enc}_j(\mathsf{pp}_j, y', m)$ を出力する．それ以外

ダミー再暗号化関数 $\mathsf{dC}^{\mathsf{REnc}}_{i.x\to j.x'}$：
入力：$\mathsf{ct}_{i.y}$
定数：$\mathsf{pp}_j,\ x,\ y'$
(1) $\mathsf{ct}_{j.y'} \leftarrow \mathsf{Enc}_j(\mathsf{pp}_j, y', \mathbf{0})$ を出力する．

図 2 ダミー再暗号化関数 $\mathsf{dC}^{\mathsf{REnc}}_{i.x\to j.x'}$
Fig. 2 Dummy Re-encryption Function $\mathsf{dC}^{\mathsf{REnc}}_{i.x\to j.x'}$

は Game 1 と同一である．以降，再暗号化オラクルにおいて，$\mathsf{P}_i(x, y_i^*) = 1$ を満たす $\mathsf{sk}_{i.x}$ が使用されることはない．

piO の正当性より，REnc の出力は，$\mathsf{Enc}_j(\mathsf{pp}_j, y', m)$ を同じ分布に従う．よって，Game 1 と Game 2 において再暗号化オラクルの出力は識別不可能なので，$|\mathsf{Win}_1 - \mathsf{Win}_2| \le \mathsf{negl}(\lambda)$ である．

Game 3. 再暗号化鍵オラクル $\mathcal{O}^{\mathsf{RKG}}(i, j, x, x')$ において，$\mathsf{P}_i(x, y_i^*) = \mathsf{P}_j(x', y_j^*) = 1$ である場合，再暗号化関数 $\mathsf{C}^{\mathsf{REnc}}_{i.x\to j.x'}$ の代わりに，図 2 に示すダミー再暗号化関数 $\mathsf{dC}^{\mathsf{REnc}}_{i.x\to j.x'}$ を用いる．それ以外は Game 2 と同じである．以降，再暗号化鍵オラクルにおいて，$\mathsf{P}_i(x, y_i^*) = 1$ を満たす $\mathsf{sk}_{i.x}$ が使用されることはない．

$\mathsf{P}_j(x', y_j^*) = 1$ より，鍵生成オラクルで $\mathsf{sk}_{j.x'}$ は入手できない．よって，Σ_j^{PE} の CPA 安全性より，$\mathsf{Enc}_j(\mathsf{pp}_j, y', m)$ と $\mathsf{Enc}_j(\mathsf{pp}_j, y', \mathbf{0})$ は識別不可能であるから，$|\mathsf{Win}_2 - \mathsf{Win}_3| \le \mathsf{negl}(\lambda)$ である．

Game 4. Challenge において生成する暗号文を暗号文空間からランダムに選ぶ．それ以外は Game 3 と同じである．このとき，b の情報が含まれていないので，明らかに $|\mathsf{Win}_4 - 1/2| = 0$ である．

また，Game 3 と Game 4 において，$\mathsf{P}_i(x, y_{i^*}^*) = 1$ を満たす $\mathsf{sk}_{i^*.x}$ を使用することはないので，$\Sigma_{i^*}^{\mathsf{PE}}$ の CPA 安全性より，2 つのゲームにおけるチャレンジ暗号文は識別不可能である．ゆえに，$|\mathsf{Win}_3 - \mathsf{Win}_4| \le \mathsf{negl}(\lambda)$ が成り立つ．

以上より，

$$\mathsf{Adv}^{\mathcal{A}}_{\mathsf{UPPRE}_1}(\lambda) = |\mathsf{Win}_0 - 1/2| = |\mathsf{Win}_1 - 1/2|$$
$$\le \sum_{t=1}^{3} |\mathsf{Win}_t - \mathsf{Win}_{t+1}| + |\mathsf{Win}_4 - 1/2| \le \mathsf{negl}(\lambda)$$

が成り立つので，定理が導かれる． □

5. GC を用いた UPPRE 方式

本節では，GC に基づく UPPRE 方式を提案する．提案構成では，GC に加えて秘密分散方式と（弱い）バッチ暗号化方式を利用する．先に再暗号化アルゴリズムの概要を述べる．

まず，秘密分散方式を用いて秘密鍵 $\mathsf{sk}_{i.x}$ を (s_1, s_2) に分割し，s_2 を再暗号化鍵に含める．s_1 を使ってバッチ暗号化の鍵ペア $(\hat{\mathsf{pk}}, \hat{\mathsf{sk}})$ を生成し，$\hat{\mathsf{sk}}$ を pp_j で暗号化したものも再暗号化鍵に加える．この再暗号化鍵から，プロキシは s_1 を入力に受け取り，内部で s_1, s_2 を使って秘密鍵を復元し，暗号文を復号する関数 $\mathsf{P}^{\mathsf{REnc}}_{i.x\to j.x'}$ を構成する．再暗号化された暗号文には，この関数を GC で変換した関数 $\bar{\mathsf{C}}$ と，そのラベルを $\hat{\mathsf{pk}}$ で暗号化したものが含まれる．受信者は，自分の秘密鍵 $\mathsf{sk}_{j.x'}$ を用いて $\hat{\mathsf{sk}}$ を入手し，ラベルの暗号化を復号する．バッチ暗号化と GC の性質より，復号したラベルを $\bar{\mathsf{C}}$ の入力として与えると $\mathsf{P}^{\mathsf{REnc}}_{i.x\to j.x'}(s_1)$ を計算することができるので，暗号文を復号することができる．

2 回目以降の再暗号化では，$\mathsf{P}^{\mathsf{REnc}}_{i.x\to j.x'}$ の代わりに，s_1 を入力にとり，$\mathsf{sk}_{i.x}$ を復元して暗号化された GC のラベルを復号する関数 $\mathsf{Q}^{\mathsf{REnc}}_{i.x\to j.x'}$ を構成する．再暗号化後の暗号文には，この関数を GC で変換した関数 $\bar{\mathsf{C}}$ が追加される．受信者は，同様に $\hat{\mathsf{sk}}$ を復号し，それを用いてラベルを復号する．l 回目の再暗号化に対応する $\bar{\mathsf{C}}_l$ にこのラベルを入力として与えると，$\mathsf{Q}^{\mathsf{REnc}}_{i.x\to j.x'}(s_1)$ が計算され，$l-1$ 回目の再暗号化に対応するラベルが復号される．$\bar{\mathsf{C}}_{l-1}, \cdots, \bar{\mathsf{C}}_1$ に同様の計算を繰り返すことで，最終的に $\mathsf{P}^{\mathsf{REnc}}_{i.x\to j.x'}(s_1)$ が計算され，平文を復号できる．

このような構成に起因して，この方式は再暗号化の度に計算すべき関数が 1 つずつ増えていく．よって，この方式はマルチホップだが，再暗号化の回数には実質的な限界がある．

5.1 提案方式の構成

$(\mathsf{Share}, \mathsf{ReCon})$ を $(2,2)$-秘密分散方式，$(\mathsf{BGen}, \mathsf{BEnc}, \mathsf{BDec})$ を弱いバッチ暗号化方式，$(\mathsf{Garble}, \mathsf{Eval})$ を GC 方式であるとする．このとき，UPPRE 方式 $\mathsf{UPPRE}_2 := (\mathsf{RKG}, \mathsf{REnc}, \mathsf{RDec})$ の構成は以下に示す通りである．

- $\mathsf{rk}_{i.x\to j.x'} \leftarrow \mathsf{RKG}(\Sigma_i^{\mathsf{PE}}, \Sigma_j^{\mathsf{PE}}, x, x', \mathsf{pp}_i, \mathsf{sk}_{i.x}, \mathsf{pp}_j)$：
  (1) $\mathsf{P}_j(x', y') = 1$ を満たす $y' \leftarrow\!\!\$\ \mathcal{Y}_j$ を選ぶ．
  (2) $(s_1, s_2) \leftarrow \mathsf{Share}(1^\lambda, \mathsf{sk}_{i.x})$ を計算する
  (3) $(\hat{\mathsf{pk}}, \hat{\mathsf{sk}}) \leftarrow \mathsf{BGen}(1^\lambda, s_1)$ を計算する．
  (4) $\tilde{\mathsf{ct}}_{j.y'} \leftarrow \mathsf{Enc}_j(\mathsf{pp}_j, y', \hat{\mathsf{sk}})$ を計算する．
  (5) $\mathsf{rk}_{i.x\to j.x'} := (\hat{\mathsf{pk}}, s_2, \tilde{\mathsf{ct}}_{j.y'}, \mathsf{pp}_i)$ を出力する．

- $\mathsf{ct}_{j.y'} \leftarrow \mathsf{REnc}(\Sigma_i^{\mathsf{PE}}, \Sigma_j^{\mathsf{PE}}, \mathsf{rk}_{i.f\to j.g}, \mathsf{ct}_{i.y})$:.
  (1) $\mathsf{ct}_{i.y}$ が Enc_i で生成されたオリジナルの暗号文ならば，図 3 に従って $\mathsf{C} := \mathsf{Q}^{\mathsf{REnc}}_{i.x\to j.x'}$ とする．
  (2) それ以外の場合，$\mathsf{ct}_{i.y} = (\hat{\mathsf{ct}}', \tilde{\mathsf{ct}}_{i.y}, \bar{\mathsf{C}}_1, \cdots, \bar{\mathsf{C}}_{l-1})$ として，図 4 に従って $\mathsf{C} := \mathsf{R}^{\mathsf{REnc}}_{i.x\to j.x'}$ とする．
  (3) $(\bar{\mathsf{C}}_l, \{\mathsf{label}_{t.b}\}_{t\in[n], b\in\{0,1\}}) \leftarrow \mathsf{Garble}(1^\lambda, \mathsf{C})$ を計算する．
  (4) $\hat{\mathsf{ct}} \leftarrow \mathsf{BEnc}(\hat{\mathsf{pk}}, \{\mathsf{label}_{t.b}\}_{t\in[n], b\in\{0,1\}})$ を計算する．

オリジナルの暗号文に対する再暗号化関数 $\mathsf{Q}^{\mathsf{REnc}}_{i.x\to j.x'}$：
入力：s_1
定数：$s_2,\ \mathsf{ct}_{i.y},\ \mathsf{pp}_i$
(1) $\mathsf{sk}'_{j.x} \leftarrow \mathsf{ReCon}(s_1, s_2)$ を計算する．
(2) $m' \leftarrow \mathsf{Dec}_i(\mathsf{pp}_i, \mathsf{sk}_{i.x}, \mathsf{ct}_{i.y})$ を出力する．

図 3 再暗号化関数 $\mathsf{Q}^{\mathsf{REnc}}_{i.x\to j.x'}$
Fig. 3 Re-encryption Function $\mathsf{Q}^{\mathsf{REnc}}_{i.x\to j.x'}$

再暗号化された暗号文に対する再暗号化関数 $\mathsf{R}^{\mathsf{REnc}}_{i.x\to j.x'}$：
入力：s_1
定数：$s_2,\ \hat{\mathsf{ct}}',\ \tilde{\mathsf{ct}}_{j.y'},\ \mathsf{pp}_i$
(1) $\mathsf{sk}'_{i.x} \leftarrow \mathsf{ReCon}(s_1, s_2)$ を計算する．
(2) $\hat{\mathsf{sk}} \leftarrow \mathsf{Dec}_i(\mathsf{pp}_i, \mathsf{sk}'_{i.x}, \tilde{\mathsf{ct}}_{i.y})$ を計算する．
(3) $\{\mathsf{label}'_t\}_{t\in[n]} \leftarrow \mathsf{BDec}(\hat{\mathsf{sk}}, \hat{\mathsf{ct}}')$ を出力する．

図 4 再暗号化関数 $\mathsf{R}^{\mathsf{REnc}}_{i.x\to j.x'}$
Fig. 4 Re-encryption Function $\mathsf{R}^{\mathsf{REnc}}_{i.x\to j.x'}$

(5) $\mathsf{ct}_{j.y'} := (\hat{\mathsf{ct}}, \tilde{\mathsf{ct}}_{j.y'}, \bar{\mathsf{C}}_1, \cdots, \bar{\mathsf{C}}_l)$ を出力する．

- $m \leftarrow \mathsf{RDec}(\mathsf{pp}_j, \mathsf{sk}_{j.x'}, \mathsf{ct}_{j.y'})$：
  (1) $\mathsf{ct}_{j.y'} = (\hat{\mathsf{ct}}, \tilde{\mathsf{ct}}_{j.y'}, \bar{\mathsf{C}}_1, \cdots, \bar{\mathsf{C}}_l)$ とする．
  (2) $\hat{\mathsf{sk}}' \leftarrow \mathsf{Dec}_j(\mathsf{pp}_j, \mathsf{sk}_{j.x'}, \tilde{\mathsf{ct}}_{j.y'})$ を計算する．
  (3) $\{\mathsf{label}'_t\}^{(l)}_{t\in[n]} \leftarrow \mathsf{BDec}(\hat{\mathsf{sk}}', \hat{\mathsf{ct}})$ を計算する．
  (4) $k = l, l-1 \cdots, 2$ について，$\{\mathsf{label}'_t\}^{(k-1)}_{t\in[n]} \leftarrow \mathsf{Eval}(\bar{\mathsf{C}}_k, \{\mathsf{label}'_t\}^{(k)}_{t\in[n]})$ を繰り返す．
  (5) $m \leftarrow \mathsf{Eval}(\bar{\mathsf{C}}_1, \{\mathsf{label}'_t\}^{(1)}_{t\in[n]})$ を出力する．

5.2 正当性

$\mathsf{ct}_{0.y_0} \leftarrow \mathsf{Enc}_0(\mathsf{pp}_0, y_0, m)$ を l 回再暗号化した暗号文がいずれも正しく復号されることを帰納法で示す．

まず，$l = 1$ の場合を考える．$(\hat{\mathsf{ct}}, \tilde{\mathsf{ct}}_{1.y_1}, \bar{\mathsf{C}}_1) \leftarrow \mathsf{REnc}(\Sigma^{\mathsf{PE}}_0, \Sigma^{\mathsf{PE}}_1, \mathsf{rk}_{0.x_0\to 1.x_1}, \mathsf{ct}_{0.x_0})$ について，$(\bar{\mathsf{C}}_1, \{\mathsf{label}_{t,b}\}_{t\in[n],b\in\{0,1\}}) \leftarrow \mathsf{Garble}(1^\lambda, \mathsf{P}^{\mathsf{REnc}}_{0.x_0\to 1.y_0})$，$\hat{\mathsf{ct}} \leftarrow \mathsf{BEnc}(\hat{\mathsf{pk}}, \{\mathsf{label}_{t,b}\}_{t\in[n],b\in\{0,1\}})$，$\tilde{\mathsf{ct}}_{1.y_1} \leftarrow \mathsf{Enc}_1(\mathsf{pp}_j, y_1, \hat{\mathsf{sk}})$ である．

$\mathsf{RDec}(\mathsf{pp}_1, \mathsf{sk}_{1.x_1}, \mathsf{ct}_{1.y_1})$ を計算すると，$\mathsf{P}_1(x_1, y_1) = 1$ だから，$\mathsf{Dec}_1(\mathsf{pp}_1, \mathsf{sk}_{1.x_1}, \tilde{\mathsf{ct}}_{1.y_1}) = \hat{\mathsf{sk}}$ である．次に，弱いバッチ暗号化方式の正当性より，$\mathsf{BDec}(\hat{\mathsf{sk}}, \hat{\mathsf{ct}}) = \{\mathsf{label}_{t,s_1[t]}\}_{t\in[n]}$ である．ここで，$s_1[t]$ は s_1 の第 t ビットである．そして，GC 方式の正当性より，$\mathsf{Eval}(\bar{\mathsf{C}}_1, \{\mathsf{label}_{t,s_1[t]}\}_{t\in[n]}) = \mathsf{Q}^{\mathsf{REnc}}_{0.x_0\to 1.x_1}(s_1)$ である．秘密分散方式の正当性より，$\mathsf{Q}^{\mathsf{REnc}}_{0.x_0\to 1.x_1}(s_1)$ は $\mathsf{sk}'_{1.x_1} \leftarrow \mathsf{ReCon}(s_1, s_2) = \mathsf{sk}_{1.x_1}$ を計算して $m' \leftarrow \mathsf{Dec}_0(\mathsf{pp}_0, \mathsf{sk}_{0.x_0}, \mathsf{ct}_{0.y_0})$ を出力する．Σ^{PE}_0 の正当性より，$m' = m$ であり，復号結果は正しい．

次に，$l > 1$ に対して，$l-1$ 回再暗号化した暗号文の復号が正当であると仮定する．このとき，$(\hat{\mathsf{ct}}, \tilde{\mathsf{ct}}_{l.y_l}, \bar{\mathsf{C}}_1, \cdots, \bar{\mathsf{C}}_l) \leftarrow \mathsf{REnc}(\Sigma^{\mathsf{PE}}_{l-1}, \Sigma^{\mathsf{PE}}_l, \mathsf{rk}_{l-1.x_{l-1}\to l.x_l}, \mathsf{ct}_{l-1.x_{l-1}})$ について，$(\bar{\mathsf{C}}_l, \{\mathsf{label}_{t,b}\}_{t\in[n],b\in\{0,1\}}) \leftarrow \mathsf{Garble}(1^\lambda, \mathsf{Q}^{\mathsf{REnc}}_{l-1.x_{l-1}\to l.x_l})$，$\hat{\mathsf{ct}} \leftarrow \mathsf{BEnc}(\hat{\mathsf{pk}}, \{\mathsf{label}_{t,b}\}_{t\in[n],b\in\{0,1\}})$，$\tilde{\mathsf{ct}}_{l.y_l} \leftarrow \mathsf{Enc}_l(\mathsf{pp}_l, y_l, \hat{\mathsf{sk}})$ である．

$\mathsf{RDec}(\mathsf{pp}_l, \mathsf{sk}_{l.x_l}, \mathsf{ct}_{l.y_l})$ を計算すると，Σ^{PE}_l の正当性より，$\mathsf{Dec}_l(\mathsf{pp}_l, \mathsf{sk}_{l.x_l}, \tilde{\mathsf{ct}}_{l.y_l}) = \hat{\mathsf{sk}}$ となる．また，弱いバッチ暗号化方式の正当性より，$\mathsf{BDec}(\hat{\mathsf{sk}}, \hat{\mathsf{ct}}) = \{\mathsf{label}_{t,s_1[t]}\}_{t\in[n]}$ である．そして，GC 方式の正当性より，$\mathsf{Eval}(\bar{\mathsf{C}}_l, \{\mathsf{label}_{t,s_1[t]}\}_{t\in[n]}) = \mathsf{Q}^{\mathsf{REnc}}_{l-1.x_{l-1}\to l.x_l}(s_1)$ である．秘密分散方式と $\Sigma^{\mathsf{PE}}_{l-1}$ の正当性より，$\mathsf{Q}^{\mathsf{REnc}}_{l-1.x_{l-1}\to l.x_l}(s_1)$ は $\mathsf{sk}'_{l-1.x_{l-1}} \leftarrow \mathsf{ReCon}(s_1, s_2) = \mathsf{sk}_{1.x_1}$ 及び $\hat{\mathsf{sk}} \leftarrow \mathsf{Dec}_{l-1}(\mathsf{pp}_{l-1}, \mathsf{sk}_{l-1.x_{l-1}}, \tilde{\mathsf{ct}}_{l-1.y_{l-1}})$ を計算する．そして，$\{\mathsf{label}'_t\}_{t\in[n]} \leftarrow \mathsf{BDec}(\hat{\mathsf{sk}}, \hat{\mathsf{ct}}')$ を出力する．ここで，$\{\mathsf{label}'_t\}_{t\in[n]}$ は $l-1$ 回目の再暗号化で計算したものであり，仮定より，これを用いて暗号文を正しく復号できる．

したがって，全ての $l \geq 1$ に対して，l 回再暗号化した暗号文を正しく復号できるので，UPPRE_2 はマルチホップ UPPRE の正当性を満たす．

5.3 安全性証明

定理 2 は提案方式 UPPRE_2 の安全性を示す．誌面の都合上，この定理の証明は概略のみを記述する．

定理 2. $N := \mathsf{poly}(\lambda)$ とする．GC 方式 $(\mathsf{Garble}, \mathsf{Eval})$，$(2,2)$-秘密分散方式 $(\mathsf{Share}, \mathsf{ReCon})$，弱いバッチ暗号化方式 (BGen, BEnc, BDec) がそれぞれ定義 2, 定義 3, 定義 4 に示す正当性と安全性を満たすとする．また，このとき，UPPRE 方式 UPPRE_2 は選択的モデルにおいて HRA 安全である．

証明. 定理 1 と同様に，チャレンジャー $\mathcal{C}$ と攻撃者 $\mathcal{A}$ の間で行われる次のゲーム列を考え，Game t において $\mathcal{A}$ が $b = b'$ である $b' \in \{0,1\}$ を出力する確率を Win_t と定める．

Game 0. 定義 8 に示す通常の安全性ゲームである．このとき $\mathsf{Adv}^{\mathcal{A}}_{\mathsf{UPPRE}_2}(\lambda) = |\mathsf{Win}_0 - 1/2|$ である．

Game 1. $\mathcal{O}^{\mathsf{REnc}}$ や Challenge において，H_{ct} に格納する値に平文 m 及び $\{\mathsf{label}_{t,s_1[t]}\}_{t\in[n]}$ を追加する．ここで，$s_1[t]$ は s_1 の第 t ビットである．それ以外は Game 0 と同一である．$\mathcal{A}$ から見て Game 0 と Game 1 は同一なので，$\mathsf{Win}_1 = \mathsf{Win}_0$ である．

Game 2. $\mathcal{O}^{\mathsf{REnc}}$ において実行する REnc のステップ 4 を変更する．BEnc の入力として $\{\mathsf{label}_{t,b}\}_{t\in[n],b\in\{0,1\}}$ の代わりに $\{(\mathsf{label}_{t,s_1[t]}, \mathsf{label}_{t,s_1[t]})\}_{t\in[n]}$ を与える．弱いバッチ暗号化方式の安全性より，$|\mathsf{Win}_1 - \mathsf{Win}_2| \leq \mathsf{negl}(\lambda)$ である．

Game 3. $\mathcal{O}^{\mathsf{REnc}}$ における REnc を変更する．ステップ 3 では，$(\bar{\mathsf{C}}_l, \{\mathsf{label}_{t,b}\}_{t\in[n],b\in\{0,1\}}) \leftarrow \mathsf{Garble}(1^\lambda, \mathsf{C})$ の代わりに $(\bar{\mathsf{C}}, \{\mathsf{label}_t\}_{t\in[n]}) \leftarrow \mathsf{Sim}(1^\lambda, \mathsf{C}(s_1))$ とする．ス

テップ 4 では，$\hat{\mathsf{ct}} \leftarrow \mathsf{BEnc}(\hat{\mathsf{pk}}, \{(\mathsf{label}_t, \mathsf{label}_t)\}_{t\in[n]})$ を計算する．GC の安全性より，$|\mathsf{Win}_2 - \mathsf{Win}_3| \leq \mathsf{negl}(\lambda)$ である．

Game 4. $\mathcal{O}^{\mathsf{REnc}}$ において実行する REnc のステップ 1,2,3 を変更する．ステップ 1 では，$\mathsf{ct}_{i.y}$ が Enc_i で生成されたオリジナルの暗号文ならば，H_{ct} から m を取り出し，$(\bar{\mathsf{C}}_l, \{\mathsf{label}_t\}_{t\in[n]}) \leftarrow \mathsf{Sim}(1^\lambda, m)$ とする．ステップ 2 では，それ以外の場合，$\mathsf{ct}_{i.y}$ を生成した再暗号化クエリで生成した $\{\mathsf{label}'_{t,s_1[t]}\}_{t\in[n]}$ を H_{ct} から取り出し，とする．$(\bar{\mathsf{C}}_l, \{\mathsf{label}_t\}_{t\in[n]}) \leftarrow \mathsf{Sim}(1^\lambda, \{\mathsf{label}'_{t,s_1[t]}\}_{t\in[n]})$ とする．そしてステップ 3 を削除する．$\bar{\mathsf{C}}_l(s_1)$ の出力は，$l = 1$ ならば m，$l > 1$ ならば $\{\mathsf{label}'_{t,s_1[t]}\}_{t\in[n]}$ であるが，これはそれぞれ $\mathsf{Q}^{\mathsf{REnc}}_{i.x\to j.x'}, \mathsf{R}^{\mathsf{REnc}}_{i.x\to j.x'}$ の出力と等しい．よって，$\mathcal{A}$ が手に入れる情報は同じだから，$\mathsf{Win}_3 = \mathsf{Win}_4$ である．
以降，再暗号化オラクルにおいて秘密鍵 $\mathsf{sk}_{i.x}$ の情報は不要になる．

Game 5. $\mathcal{O}^{\mathsf{RKG}}(i, j, x, x')$ について，$\mathsf{P}_i(x, y_i^*) = 1$ かつ $\mathsf{P}_j(x', y_j^*) = 1$ であるときの RKG の処理を変更する．ステップ 4 において，$\tilde{ct}_{j.y'} \leftarrow \mathsf{Enc}_j(\mathsf{pp}_j, y', \hat{\mathsf{sk}})$ の代わりに $\tilde{ct}_{j.y'} \leftarrow \mathsf{Enc}_j(\mathsf{pp}_j, y', \mathbf{0})$ を計算する．Σ_j^{PE} の CPA 安全性より，$|\mathsf{Win}_4 - \mathsf{Win}_5| \leq \mathsf{negl}(\lambda)$ である．

Game 6. $\mathcal{O}^{\mathsf{RKG}}(i, j, x, x')$ について，$\mathsf{P}_i(x, y_i^*) = 1$ かつ $\mathsf{P}_j(x', y_j^*) = 1$ であるときの RKG の処理を変更する．ステップ 3 において，$(\hat{\mathsf{pk}}, \hat{\mathsf{sk}}) \leftarrow \mathsf{BGen}(1^\lambda, s_1)$ の代わりに $(\hat{\mathsf{pk}}, \hat{\mathsf{sk}}) \leftarrow \mathsf{BGen}(1^\lambda, \mathbf{0})$ を計算する．Game 5 の変更により，$\hat{\mathsf{sk}}$ を使用することはないので，弱いバッチ暗号化方式の安全性に帰着することができる．よって，$|\mathsf{Win}_5 - \mathsf{Win}_6| \leq \mathsf{negl}(\lambda)$ が成り立つ．

Game 7. $\mathcal{O}^{\mathsf{RKG}}(i, j, x, x')$ について，$\mathsf{P}_i(x, y_i^*) = 1$ かつ $\mathsf{P}_j(x', y_j^*) = 1$ であるときの RKG の処理を変更する．ステップ 2 において，$(s_1, s_2) \leftarrow \mathsf{Share}(1^\lambda, \mathsf{sk}_{i.x})$ の代わりに $(s_1, s_2) \leftarrow \mathsf{Share}(1^\lambda, \mathbf{0})$ を計算する．Game 6 の変更によって，s_1 を使用することはないので，秘密分散方式の安全性より，$|\mathsf{Win}_6 - \mathsf{Win}_7| \leq \mathsf{negl}(\lambda)$ が成り立つ．
これ以降，$\mathsf{P}_i(x, y_i^*) = 1$ を満たすような $\mathsf{sk}_{i.x}$ は使用されない．よって，Σ_i^{PE} の CPA 安全性より，$|\mathsf{Win}_7 - 1/2| \leq \mathsf{negl}(\lambda)$ である．

以上より，

$$\mathsf{Adv}^{\mathcal{A}}_{\mathsf{UPPRE}_2}(\lambda) = |\mathsf{Win}_0 - 1/2| = |\mathsf{Win}_1 - 1/2|$$
$$\leq \sum_{t=1}^{6} |\mathsf{Win}_t - \mathsf{Win}_{t+1}| + |\mathsf{Win}_7 - 1/2| \leq \mathsf{negl}(\lambda)$$

が成り立つので，定理が導かれる． □

6. おわりに

本論文では，UPRE を述語暗号に拡張した．具体的には，述語暗号に対する UPRE のモデルと HRA 安全性を定式化し，2 つの一般的構成を提案した．1 つ目の方式は piO，2 つ目の方式は GC から構成され，いずれの方式も定式化した HRA 安全性を達成する．これらの一般的構成の instantiation を示すことは今後の課題である．

謝辞 本研究の一部は JSPS 科研費 JP23K24846 の助成を受けたものです．また，本研究の一部は，JST 経済安全保障重要技術育成プログラム【JPMJKP24U2】の支援を受けたものです．

参考文献

[1] Giuseppe Ateniese, Karyn Benson, and Susan Hohenberger: Key-Private Proxy Re-encryption, CT-RSA 2009, LNCS, Vol. 5473, pp. 279–294. Springer (2009).
[2] Giuseppe Ateniese, Kevin Fu, Matthew Green, and Susan Hohenberger: Improved Proxy Re-Encryption Schemes with Applications to Secure Distributed Storage, NDSS 2005, The Internet Society (2005).
[3] Matt Blaze, Gerrit Bleumer, and Martin Strauss. Divertible protocols and atomic proxy cryptography, EUROCRYPT 1998, LNCS, Vol. 1403, pp. 127–144. Springer (1998).
[4] Dan Boneh and Matthew Franklin: Identity Based Encryption From thr Weil Pairing, CRYPTO 2001, LNCS, Vol. 2139, pp. 213–229. Springer (2001).
[5] Aloni Cohen: What about bob? The inadequacy of CPA security for proxy reencryption, PKC 2019, LNCS, Vol. 11443, pp. 287–316. Springer (2019).
[6] Ran Canetti and Susan Hohenberger: Chosen-ciphertext secure proxy re-encryption, CCS 2007, pp. 185–194. ACM (2007).
[7] Ran Canetti, Huijia Lin, Stefano Tessaro and Vinod Vaikuntanathan: Obfuscation of probabilistic circuits and applications, TCC 2015, LNCS, Vol. 9015, pp. 468–497. Springer (2015).
[8] Nico Döttling and Ryo Nishimaki: Universal Proxy Re-Encryption, PKC 2021, LNCS, Vol. 12710, pp. 512–542. Springer (2021).
[9] Georg Fuchsbauer, Chethan Kamath, Karen Klein, and Krzysztof Pietrzak: Adaptively Secure Proxy Re-encryption, PKC 2019, LNCS, Vol. 11443, pp. 317–346. Springer (2019)
[10] Shuichi Katsumata, Ryo Nishimaki, Shota Yamada, and Takashi Yamakawa: Adaptively secure inner product encryption from LWE, ASIACRYPT 2020, LNCS, Vol. 12493, pp. 375–404. Springer (2020).
[11] Jonathan Katz, Amit Sahai and Brent Waters: Predicate Encryption Supporting Disjunctions, Polynomial Equations, and Inner Products, EUROCRYPT 2008, LNCS, Vol. 4965, pp. 146–162. Springer (2008).
[12] Amit Sahai and Brent Waters: Fuzzy Identity-Based Encryption, EUROCRYPT 2005, LNCS, Vol. 3494, pp. 457–473. Springer (2005).