公開鍵暗号における送受信者編集否認可能性

豊岡 叶望[1,a)]　渡邉 洋平[1,2]　岩本 貢[1]

概要：否認可能暗号（Deniable Encryption: DE）とは，公開された暗号文に対応する平文を開示するよう強制されたあるユーザが，暗号文の中身について嘘をつくことを可能にする暗号方式であり，プライバシー保護の観点から重要な技術である．DE には，送信者が暗号文に対応する偽の平文と偽の乱数を生成することを可能にする送信者否認可能性，受信者が暗号文を偽の平文に復号できるような偽の秘密鍵を生成することを可能にする受信者否認可能性，送信者否認と受信者否認の双方が可能な送受信者否認可能性が存在する．また，鍵生成や暗号化に別のアルゴリズムの使用を許容する Flexible モデル及び，別のアルゴリズムの使用を許容しない Full モデルという分類も存在し，達成する安全性と使用するモデルにより，様々な構成が提案されている．（受信者）編集否認可能公開鍵暗号（Goldwasser et al., TCC 2017）は，Flexible モデルの受信者否認可能公開鍵暗号の亜種であり，暗号文を，元の平文を編集関数に通して得られる平文に復号できるような偽の秘密鍵を生成することを可能にする．本研究では，送信者側での編集否認可能性についての検討を行い，受信者側は Flexible モデル，送信者側は Full モデルでの送受信者編集否認可能公開鍵暗号の定義と構成法を示す．

キーワード：編集否認可能公開鍵暗号，否認可能暗号，関数型暗号

Edit-Bideniability for Public-Key Encryption

Towa Toyooka[1,a)]　Yohei Watanabe[1,2]　Mitsugu Iwamoto[1]

Abstract: Deniable Encryption (DE) is a cryptographic scheme that enables a user, who may be coerced to reveal the messages corresponding to the user's public ciphertexts, to lie about which messages the user encrypted, and is an important technology from the perspective of privacy-preserving technologies. There are several types of deniability: sender-deniability, which allows the sender to generate a fake message and fake randomness for a ciphertext; receiver-deniability, which allows the receiver to generate a fake secret key that decrypts the ciphertext to a fake message; and bi-deniability, which combines both sender- and receiver-deniability. Furthermore, DE schemes are classified into the flexible model, which permits the use of different algorithms for key generation or encryption, and the full model, which does not. Various constructions have been proposed depending on the desired security level and the model used. (Receiver) Edit-Deniable Public Key Encryption (Goldwasser et al., TCC 2017) is a variant of receiver-deniable public key encryption in the flexible model. It enables the generation of a fake secret key that decrypts a ciphertext to a message obtained by applying an edit function to the original message. In this research, we investigate sender-side edit-deniability. We define and construct an edit-bideniable public key encryption scheme where the receiver-deniability is in the flexible model and the sender-deniability is in the full model.

Keywords: Edit-Deniable Public-Key Encryption, Deniable Encryption, Functional Encryption

1. はじめに

1.1 背景

否認可能暗号（Deniable Encryption: DE）[8] とは，公

[1] 電気通信大学 / The University of Electro-Communications
[2] 国立研究開発法人 産業技術総合研究所 / AIST
[a)] t.toyooka@uec.ac.jp

開された暗号文に対応する平文を開示するよう強制されたあるユーザが，暗号文の中身について嘘をつくことを可能にする暗号方式であり，プライバシに配慮したデータの利活用という面において重要な技術である．DE には，送信者が暗号文に対応する偽の平文と偽の乱数を生成することを可能にする送信者否認可能性，受信者が暗号文を偽の平文に復号できるような偽の秘密鍵を生成することを可能にする受信者否認可能性，送信者否認と受信者否認の双方が可能な送受信者否認可能性が存在する．また，鍵生成や暗号化の際に別のアルゴリズムの使用を許容する Flexible モデル及び，別のアルゴリズムの使用を許容しない Full モデルという分類も存在し，達成する安全性と使用するモデルにより，様々な構成が提案されている．なお，別のアルゴリズムの使用を許容するモデルには，Flexible モデル [3], [8]，Multi-Distributional モデル [5], [9], [10], [18]，Dual-Key/Scheme モデル [13]，Weak モデル [1], [2] といったいくつかの名付け方のバリエーションが存在するが，本稿では Flexible モデルで統一する．

Goldwasser ら [13] によって提案された（受信者）編集否認可能公開鍵暗号は，Flexible モデルの受信者否認可能公開鍵暗号の亜種であり，暗号分 ct を，もとの平文 m を編集記述 e に基づいて編集した平文 $m^* = \mathsf{Edit}(m, e)$ に復号することのできる偽の秘密鍵 $\mathsf{sk}_{\mathsf{ct},e}$ を生成することのできる暗号方式である．Goldwasser ら [13] は，秘密鍵のサイズが編集記述 e のサイズに線形になるような（受信者）編集否認可能公開鍵暗号の具体的構成を，公開鍵暗号と Garbled Circuit [21] を用いて構成している．

1.2 本稿の貢献

Goldwasser ら [13] による編集否認可能公開鍵暗号の構成では，受信者否認のみが可能であり，送信者側が暗号文の中身の開示の強制を受けたとき，送信者側が否認することは不可能である．そこで本研究では，編集否認可能公開鍵暗号の送信者否認可能性，及び送受信者否認可能性への拡張を行う．はじめに送信者否認可能 Single-Key Public-Key Functional Encryption (Single-Key PKFE) という暗号方式の定義，及び公開鍵暗号，送信者否認可能公開鍵暗号，Garbled Circuit [21] を用いた具体的構成法を提案する．次に送受信者編集否認可能公開鍵暗号の定義を提案し，送信者否認可能 Single-Key PKFE を用いた送受信者編集否認可能公開鍵暗号の具体的構成法の提案を行う．

1.3 本稿の構成

本論文の構成は以下の通りである．第 2 節で記法と Garbled Circuit 及び送信者否認可能公開鍵暗号のモデル，各種定義を確認する．第 3 節では，送信者否認可能 Single-Key PKFE の定義と構成法の提案を行う．第 4 節では，送受信者編集否認可能公開鍵暗号の定義と構成法の提案を行う．最後に第 5 節でまとめと今後の課題を述べる．

2. 準備

2.1 記法

確率的多項式時間（Probabilistic Polynomial-Time：PPT）アルゴリズム A に x を入力して y を出力することを $y \leftarrow A(x)$ と表記し，A 中の内部乱数 r を明記する場合は $y \leftarrow A(x; r)$ とする．λ に関する正の多項式を $\mathsf{poly}(\lambda)$ と表記する．$n \in \mathbb{N}$ に対して $[n] \coloneqq \{1, 2, \ldots, n\}$ とし，ビット列 $x, y \in \{0,1\}^*$ に対して，x の i 番目のビットを $x[i]$ と表記し，x と y の結合を $x \| y$ と表記する．また，集合 A から一様に値 x を選ぶことを $x \xleftarrow{\$} A$ と表記する．

定義 2.1 (無視可能関数)．関数 $\mathsf{negl}\colon \mathbb{N} \to \mathbb{R}_{\geq 0}$ が無視可能関数であるとは，任意の $c > 0$ に対してある $n \in \mathbb{N}$ が存在し，任意の $k > n$ に対して $\mathsf{negl}(k) < k^{-c}$ が成立することをいう．

2.2 Garbled Circuits

定義 2.2 (Garbling 方式 [4])．回路族 $\mathcal{C} = \{C\colon \{0,1\}^{n_1(\lambda)} \to \{0,1\}^{n_2(\lambda)}\}$ に対する Garbling 方式 GC は以下の 4 つの PPT アルゴリズム $(\mathsf{Setup}, \mathsf{Garble}, \mathsf{Encode}, \mathsf{Eval})$ からなる．

- $\underline{\mathsf{Setup}(1^\lambda) \to \mathbf{K}}$：セキュリティパラメータ λ を入力にとり，ラベル $\mathbf{K} = (k_{i,b})_{i \in [n_1(\lambda)], b \in \{0,1\}}$ を出力する．
- $\underline{\mathsf{Garble}(C, \mathbf{K}) \to \widehat{C}}$：回路 C とラベル $\mathbf{K}$ を入力にとり，garbled 回路 $\widehat{C}$ を出力する．
- $\underline{\mathsf{Encode}(x, \mathbf{K}) \to \mathbf{K}_x}$：回路の入力 $x \in \{0,1\}^{n_1(\lambda)}$ とラベル $\mathbf{K}$ を入力にとり，入力ラベル $\mathbf{K}_x = (k_{i,x[i]})_{i \in [n_1(\lambda)]}$ を出力する．
- $\underline{\mathsf{Eval}(\widehat{C}, \mathbf{K}_x) \to C(x)}$：garbled 回路 $\widehat{C}$ と入力ラベル $\mathbf{K}_x$ を入力にとり，回路の評価結果 $C(x)$ を出力する．

Garbling 方式の正当性と安全性の定義は以下の通りである．

定義 2.3 (正当性)．$\mathsf{GC} = (\mathsf{Setup}, \mathsf{Garble}, \mathsf{Encode}, \mathsf{Eval})$ が正当性を満たすとは，任意のセキュリティパラメータ λ，回路 $C\colon \{0,1\}^{n_1(\lambda)} \to \{0,1\}^{n_2(\lambda)}$，入力 $x \in \{0,1\}^{n_1(\lambda)}$ に対して以下が成り立つことをいう．

$$\Pr\left[C(x) = \mathsf{Eval}(\widehat{C}, \mathbf{K}_x) : \begin{array}{r} \mathbf{K} \leftarrow \mathsf{Setup}(1^\lambda) \\ \widehat{C} \leftarrow \mathsf{Garble}(C, \mathbf{K}) \\ \mathbf{K}_x \leftarrow \mathsf{Encode}(x, \mathbf{K}) \end{array}\right] = 1.$$

定義 2.4 (安全性)．$\mathsf{GC} = (\mathsf{Setup}, \mathsf{Garble}, \mathsf{Encode}, \mathsf{Eval})$ が安全性を満たすとは，任意のセキュリティパラメータ λ，回路 $C\colon \{0,1\}^{n_1(\lambda)} \to \{0,1\}^{n_2(\lambda)}$，$C(x_0) = C(x_1)$ を満たす入力 $x_0, x_1 \in \{0,1\}^{n_1(\lambda)}$ に対して，$(\widehat{C}, \mathbf{K}_{x_0})$ と $(\widehat{C}, \mathbf{K}_{x_1})$ が計算量的に識別不可能であることをいう．ただし，$\mathbf{K} \leftarrow \mathsf{Setup}(1^\lambda), \widehat{C} \leftarrow \mathsf{Garble}(C, \mathbf{K}), \mathbf{K}_{x_0} \leftarrow \mathsf{Encode}(x_0, \mathbf{K}), \mathbf{K}_{x_1} \leftarrow \mathsf{Encode}(x_1, \mathbf{K})$ である．

なお，2 入力回路族 $\mathcal{C} = \{C\colon \{0,1\}^{n_1(\lambda)} \times \{0,1\}^{n_2(\lambda)} \to \{0,1\}^{n_3(\lambda)}\}$ に対する Garbling 方式では，$(\mathbf{K}^1, \mathbf{K}^2) \leftarrow \mathsf{GC.Setup}(1^\lambda)$ でラベル $\mathbf{K}^1 = (k^1_{i,b})_{i\in[n_1(\lambda)],b\in\{0,1\}}, \mathbf{K}^2 = (k^2_{i,b})_{i\in[n_2(\lambda)],b\in\{0,1\}}$ を生成し，2 つの入力 $x \in \{0,1\}^{n_1(\lambda)}$, $y \in \{0,1\}^{n_2(\lambda)}$ それぞれに対して $\mathbf{K}^1_x \leftarrow \mathsf{GC.Encode}(x, \mathbf{K}^1)$, $\mathbf{K}^2_y \leftarrow \mathsf{GC.Encode}(y, \mathbf{K}^2)$ で入力ラベル $\mathbf{K}^1_x, \mathbf{K}^2_y$ を生成して，回路の評価値 $C(x,y)$ を $C(x,y) \leftarrow \mathsf{GC.Eval}(\widehat{C}, (\mathbf{K}^1_x, \mathbf{K}^2_y))$ で得るものとする．

2.3 送信者否認可能公開鍵暗号

送信者否認可能公開鍵暗号方式とは，任意の平文 m, m^* に対して，$\mathsf{ct} = \mathsf{Enc}(\mathsf{pk}, m; r) = \mathsf{Enc}(\mathsf{pk}, m^*; r^*)$ となるような偽の乱数 r^* を送信者が生成可能な公開鍵暗号方式である．以下に送信者が Full モデルにおける送信者否認可能公開鍵暗号方式の定義を示す．

定義 2.5 (送信者否認可能公開鍵暗号)．平文空間 $\mathcal{M}$，乱数空間 $\mathcal{R}$ に対する送信者否認可能公開鍵暗号方式 SDE は以下の 4 つの PPT アルゴリズム $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{SFake})$ からなる．

- $\underline{\mathsf{Gen}(1^\lambda) \to (\mathsf{pk}, \mathsf{sk})}$：セキュリティパラメータ λ を入力にとり，公開鍵 pk と秘密鍵 sk を出力する．
- $\underline{\mathsf{Enc}(\mathsf{pk}, m; r) \to \mathsf{ct}}$：公開鍵 pk と平文 $m \in \mathcal{M}$ を入力にとり，乱数 $r \in \mathcal{R}$ を使用して暗号文 ct を出力する．
- $\underline{\mathsf{Dec}(\mathsf{sk}, \mathsf{ct}) \to m}$：秘密鍵 sk と暗号文 ct を入力にとり，平文 $m \in \mathcal{M}$ を出力する．
- $\underline{\mathsf{SFake}(\mathsf{pk}, m, r, m^*) \to r^*}$：公開鍵 pk，元の平文 $m \in \mathcal{M}$，暗号化に使用した乱数 $r \in \mathcal{R}$，偽の平文 $m^* \in \mathcal{M}$ を入力にとり，偽の乱数 $r^* \in \mathcal{R}$ を出力する．

送信者否認可能公開鍵暗号方式の正当性，IND-CPA 安全性，送信者否認可能性の定義は以下の通りである．

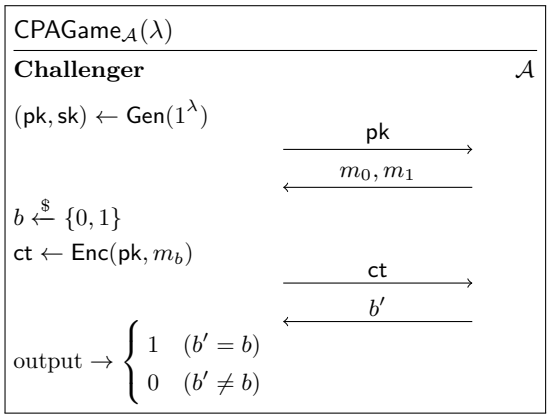
定義 2.6 (正当性)．$\mathsf{SDE} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{SFake})$ が正当性を満たすとは，任意のセキュリティパラメータ λ，平文 $m \in \mathcal{M}$ に対して以下が成り立つことをいう．

$$\Pr\left[\mathsf{Dec}(\mathsf{sk}, \mathsf{ct}) = m : \begin{matrix}(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}(1^\lambda) \\ \mathsf{ct} \leftarrow \mathsf{Enc}(\mathsf{pk}, m; r)\end{matrix}\right] = 1 - \mathsf{negl}(\lambda).$$

定義 2.7 (IND-CPA 安全性)．$\mathsf{SDE} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{SFake})$ が IND-CPA 安全性を満たすとは，任意の PPT 攻撃者 $\mathcal{A}$ に対して以下が成り立つことをいう．

$$\left|\Pr[\mathsf{CPAGame}_{\mathcal{A}}(\lambda) \to 1] - \frac{1}{2}\right| \le \mathsf{negl}(\lambda).$$

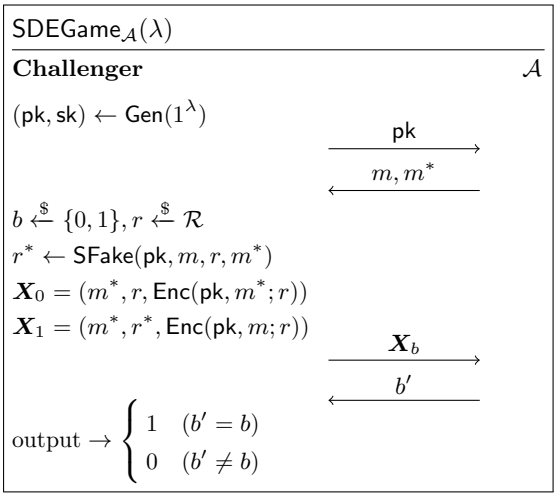
ただし，$\mathsf{CPAGame}_{\mathcal{A}}(\lambda)$ は PPT 攻撃者 $\mathcal{A}$ と Challenger 間のゲームであって，次のように定義される．



定義 2.8 (送信者否認可能性)．$\mathsf{SDE} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{SFake})$ が $\delta(\lambda)$-送信者否認可能性を満たすとは，任意の PPT 攻撃者 $\mathcal{A}$ に対して以下が成り立つことをいう．

$$\left|\Pr[\mathsf{SDEGame}_{\mathcal{A}}(\lambda) \to 1] - \frac{1}{2}\right| \le \delta(\lambda).$$

ただし，$\mathsf{SDEGame}_{\mathcal{A}}(\lambda)$ は PPT 攻撃者 $\mathcal{A}$ と Challenger 間のゲームであって，次のように定義される．



3. Single-Key Public-Key Functional Encryption

本節では，Single-Key Public-Key Functional Encryption (Singel-Key PKFE) [12], [14], [20] について述べる．第 3.1 節では，Goldwasser ら [13] による受信者編集否認可能公開鍵暗号の構成に用いられる Single-Key PKFE の定義を示す．第 3.2 節では，送信者否認可能 Single-Key PKFE を新たに定義し，その構成法を示す．

3.1 Single-Key Public-Key Functional Encryption with Special Decryption

Single-Key PKFE とは，関数型暗号（Functional Encryption: FE）[6], [17] の一種であり，安全性ゲームにおける攻撃者の関数秘密鍵クエリの回数が 1 回に制限された公開鍵型の FE である [12], [14], [20]．特に本節では，適応的安全，すなわち攻撃者が関数鍵クエリを行ったあとに平文を

選択できるような安全性を満たす方式について述べる．

定義 3.1 (Single-Key PKFE)．関数 $F\colon \{0,1\}^{n_1(\lambda)} \times \{0,1\}^{n_2(\lambda)} \to \{0,1\}^{n_3(\lambda)}$ に対する Single-Key Public-Key Functional Encryption 方式 FE は以下の 4 つの PPT アルゴリズム $(\mathsf{Setup}, \mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ からなる．

- $\underline{\mathsf{Setup}(1^\lambda) \to (\mathsf{mpk}, \mathsf{msk})}$：セキュリティパラメータ λ を入力にとり，マスター公開鍵 mpk とマスター秘密鍵 msk を出力する．
- $\underline{\mathsf{Gen}(\mathsf{msk}, y) \to \mathsf{sk}_y}$：マスター秘密鍵 msk と $y \in \{0,1\}^{n_2(\lambda)}$ を入力に取り，秘密鍵 sk_y を出力する．
- $\underline{\mathsf{Enc}(\mathsf{mpk}, x) \to \mathsf{ct}}$：マスター公開鍵 mpk と平文 $x \in \{0,1\}^{n_1(\lambda)}$ を入力にとり，暗号文 ct を出力する．
- $\underline{\mathsf{Dec}(\mathsf{sk}_y, \mathsf{ct}) \to F(x,y)}$：秘密鍵 sk_y と暗号文 ct を入力にとり，関数の評価値 $F(x,y)$ を出力する．

Single-Key PKFE 方式の正当性と安全性の定義は以下の通りである．

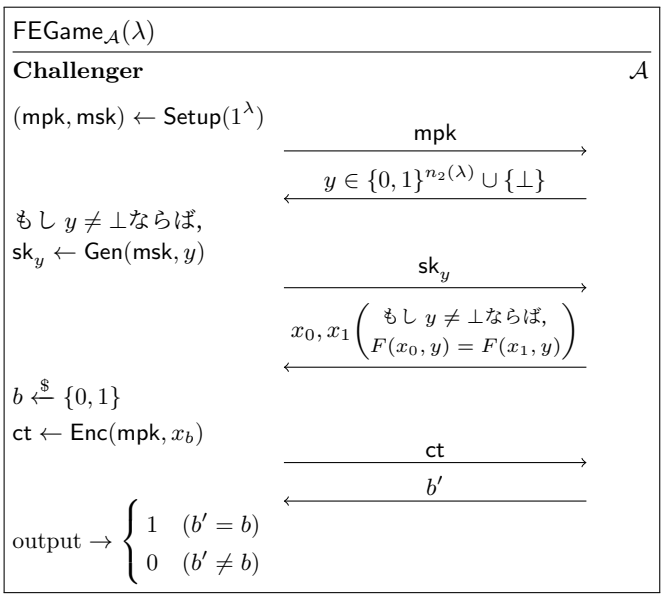
定義 3.2 (正当性)．$\mathsf{FE} = (\mathsf{Setup}, \mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ が正当性を満たすとは，任意のセキュリティパラメータ λ，$x \in \{0,1\}^{n_1(\lambda)}$，$y \in \{0,1\}^{n_2(\lambda)}$ に対して以下が成り立つことをいう．

$$\Pr\left[F(x,y) = \mathsf{Dec}(\mathsf{sk}_y, \mathsf{ct}) : \begin{array}{c}(\mathsf{mpk}, \mathsf{msk}) \leftarrow \mathsf{Setup}(1^\lambda) \\ \mathsf{sk}_y \leftarrow \mathsf{Gen}(\mathsf{msk}, y) \\ \mathsf{ct} \leftarrow \mathsf{Enc}(\mathsf{mpk}, x)\end{array}\right] = 1.$$

定義 3.3 (安全性)．$\mathsf{FE} = (\mathsf{Setup}, \mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ が安全性を満たすとは，任意の PPT 攻撃者 $\mathcal{A}$ に対して以下が成り立つことをいう．

$$\left|\Pr[\mathsf{FEGame}_{\mathcal{A}}(\lambda) \to 1] - \frac{1}{2}\right| \leq \mathsf{negl}(\lambda).$$

ただし，$\mathsf{FEGame}_{\mathcal{A}}(\lambda)$ は PPT 攻撃者 $\mathcal{A}$ と Challenger 間のゲームであって，次のように定義される．



さらに [13] では，Single-Key PKFE に対して，以下の Special Decryption と呼ばれる追加の性質を要求する．

定義 3.4 (Special Decryption)．$\mathsf{FE} = (\mathsf{Setup}, \mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ が Special Decryption をもつとは，任意のセキュリティパラメータ λ，$x \in \{0,1\}^{n_1(\lambda)}$ に対して以下が成り立つように復号アルゴリズム Dec を拡張できることをいう．

$$\Pr\left[\mathsf{Dec}(\mathsf{msk}, \mathsf{ct}) = x : \begin{array}{c}(\mathsf{mpk}, \mathsf{msk}) \leftarrow \mathsf{Setup}(1^\lambda) \\ \mathsf{ct} \leftarrow \mathsf{Enc}(\mathsf{mpk}, x)\end{array}\right] = 1.$$

3.2 Single-Key PKFE における送信者否認可能性

送信者否認可能 Single-Key PKFE を以下のように定義する．

定義 3.5 (送信者否認可能 Single-Key PKFE)．関数 $F\colon \{0,1\}^{n_1(\lambda)} \times \{0,1\}^{n_2(\lambda)} \to \{0,1\}^{n_3(\lambda)}$ に対する送信者否認可能 Single-Key Public-Key Functional Encryption 方式 SDFE は以下の 5 つの PPT アルゴリズム $(\mathsf{Setup}, \mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{SFake})$ からなる．

- $\underline{\mathsf{Setup}(1^\lambda) \to (\mathsf{mpk}, \mathsf{msk})}$：セキュリティパラメータ λ を入力にとり，マスター公開鍵 mpk とマスター秘密鍵 msk を出力する．
- $\underline{\mathsf{Gen}(\mathsf{msk}, y) \to \mathsf{sk}_y}$：マスター秘密鍵 msk と $y \in \{0,1\}^{n_2(\lambda)}$ を入力に取り，秘密鍵 sk_y を出力する．
- $\underline{\mathsf{Enc}(\mathsf{mpk}, x) \to \mathsf{ct}}$：マスター公開鍵 mpk と平文 $x \in \{0,1\}^{n_1(\lambda)}$ を入力にとり，暗号文 ct を出力する．
- $\underline{\mathsf{Dec}(\mathsf{sk}_y, \mathsf{ct}) \to F(x,y)}$：秘密鍵 sk_y と暗号文 ct を入力にとり，関数の評価値 $F(x,y)$ を出力する．
- $\underline{\mathsf{SFake}(\mathsf{pk}, m, r, m^*) \to r^*}$：公開鍵 pk，元の平文 $m \in \mathcal{M}$，暗号化に使用した乱数 $r \in \mathcal{R}$，偽の平文 $m^* \in \mathcal{M}$ を入力にとり，偽の乱数 $r^* \in \mathcal{R}$ を出力する．

送信者否認可能 Single-Key PKFE 方式の正当性，安全性，送信者否認可能性の定義は，定義 3.2，定義 3.3，定義 2.8 とそれぞれ同様である．.

Goldwasser らによる Single-Key PKFE の構成（[13], Construction B.1）をもとに，関数 $F\colon \{0,1\}^{n_1(\lambda)} \times \{0,1\}^{n_2(\lambda)} \to \{0,1\}^{n_3(\lambda)}$ に対する送信者否認可能 Single-Key PKFE の構成を提案する．なお，構成には公開鍵暗号方式 $\mathsf{PKE} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$，送信者否認可能公開鍵暗号方式 $\mathsf{SDE} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{SFake})$，回路族 $\mathcal{C} = \{C\colon \{0,1\}^{n_1(\lambda)+n_2(\lambda)} \to \{0,1\}^{n_3(\lambda)}\}$ に対する Garbling 方式 $\mathsf{GC} = (\mathsf{Setup}, \mathsf{Garble}, \mathsf{Encode}, \mathsf{Eval})$ を用いる．

構成 3.1. $\underline{\mathsf{SDFE.Setup}(1^\lambda) \to (\mathsf{mpk}, \mathsf{msk})}$：セキュリティパラメータ λ を入力にとり，以下を実行する．

1. 任意の $i \in [n_2(\lambda)], b \in \{0,1\}$ に対して，$(\mathsf{sk}_{i,b}, \mathsf{pk}_{i,b}) \leftarrow \mathsf{PKE.Gen}(1^\lambda)$.
2. $(\mathsf{dsk}, \mathsf{dpk}) \leftarrow \mathsf{SDE.Gen}(1^\lambda)$.
3. $\mathsf{mpk} := \big((\mathsf{pk}_{i,b})_{i \in [n_2(\lambda)], b \in \{0,1\}}, \mathsf{dpk}\big)$ と $\mathsf{msk} :=$

$\left((\mathsf{sk}_{i,b})_{i\in[n_2(\lambda)],b\in\{0,1\}}, \mathsf{dsk}\right)$ を出力．

$\underline{\mathsf{SDFE.Gen}(\mathsf{msk}, y) \to \mathsf{sk}_y}$：マスター秘密鍵 msk と $y \in \{0,1\}^{n_2(\lambda)}$ を入力に取り，以下を実行する．

1. $\mathsf{sk}_y \coloneqq (\mathsf{sk}_{i,y[i]})_{i\in[n_2(\lambda)]}$ を出力．

$\underline{\mathsf{SDFE.Enc}(\mathsf{mpk}, x; r) \to \mathsf{ct}}$：マスター公開鍵 mpk と平文 $x \in \{0,1\}^{n_1(\lambda)}$ を入力にとり，以下を実行する．

1. $(\mathbf{K}^1, \mathbf{K}^2) \leftarrow \mathsf{GC.Setup}(1^\lambda; r_{\mathsf{Setup}})$.
2. $\widehat{F} \leftarrow \mathsf{GC.Garble}(F, (\mathbf{K}^1, \mathbf{K}^2); r_{\mathsf{Garble}})$.
3. $\mathbf{K}^1_x = \mathsf{GC.Encode}(x, \mathbf{K}^1)$.
4. $\widetilde{\mathbf{K}^1_x} \leftarrow \mathsf{SDE.Enc}(\mathsf{dpk}, \mathbf{K}^1_x; r_{\mathsf{Enc}})$.
5. 任意の $i \in [n_2(\lambda)], b \in \{0,1\}$ に対して，$c_{i,b} \leftarrow \mathsf{PKE.Enc}(\mathsf{pk}_{i,b}, k^2_{i,b}; r_{i,b})$.
6. $\mathsf{ct} \coloneqq \left(\widehat{F}, \widetilde{\mathbf{K}^1_x}, (c_{i,b})_{i\in[n_2(\lambda)],b\in\{0,1\}}\right)$ を出力．

$\underline{\mathsf{SDFE.Dec}(\mathsf{sk}_y, \mathsf{ct}) \to F(x,y)}$：秘密鍵 sk_y と暗号文 ct を入力にとり，以下を実行する．

1. $\mathbf{K}^1_x = \mathsf{SDE.Dec}(\mathsf{dsk}, \widetilde{\mathbf{K}^1_x})$.
2. 任意の $i \in [n_2(\lambda)]$ に対して，$k^2_{i,y[i]} = \mathsf{PKE.Dec}(\mathsf{sk}_{i,y[i]}, c_{i,y[i]})$.
3. $\mathbf{K}^2_y \coloneqq (k^2_{i,y[i]})_{i\in[n_2(\lambda)]}$.
4. $F(x,y) = \mathsf{GC.Eval}(\widehat{F}, (\mathbf{K}^1_x, \mathbf{K}^2_y))$.

$\underline{\mathsf{SDFE.SFake}(\mathsf{pk}, x, r, x^*) \to r^*}$：公開鍵 pk，元の平文 $m \in \mathcal{M}$，暗号化に使用した乱数 $r = \left(r_{\mathsf{Setup}}, r_{\mathsf{Garble}}, r_{\mathsf{Enc}}, (r_{i,b})_{i\in[n_2(\lambda)],b\in\{0,1\}}\right)$，偽の平文 $m^* \in \mathcal{M}$ を入力にとり，以下を実行する．

1. $r^*_{\mathsf{Enc}} \leftarrow \mathsf{SDE.SFake}(\mathsf{dpk}, \mathbf{K}^1_x, r_{\mathsf{Enc}}, \mathbf{K}^1_{x^*})$.
2. $r^* \coloneqq \left(r_{\mathsf{Setup}}, r_{\mathsf{Garble}}, r^*_{\mathsf{Enc}}, (r_{i,b})_{i\in[n_2(\lambda)],b\in\{0,1\}}\right)$ を出力．

定理 3.1. 構成 3.1 の SDFE は正当性を満たす．

証明. PKE, SDE, GC のそれぞれの正当性より成立． □

定理 3.2. 構成 3.1 の SDFE は安全性を満たす．

証明. PKE, SDE の IND-CPA 安全性及び，GC の安全性より成立． □

定理 3.3. SDE が $\delta(\lambda)$-送信者否認可能であるとき，構成 3.1 の SDFE は $\delta(\lambda)$-送信者否認可能である．

証明. SDFE.SFake で出力される偽の乱数 $r^* \coloneqq \left(r_{\mathsf{Setup}}, r_{\mathsf{Garble}}, r^*_{\mathsf{Enc}}, (r_{i,b})_{i\in[n_2(\lambda)],b\in\{0,1\}}\right)$ のうち，SDFE の送信者否認可能性に関わるのは r^*_{Enc} のみであり，r^*_{Enc} は SDE.SFake によって生成される．よって，SDE が $\delta(\lambda)$-送信者否認可能であるならば，構成 3.1 の SDFE は $\delta(\lambda)$-送信者否認可能である． □

Special Decryption については以下が成立する．

定理 3.4. $F\colon \{0,1\}^{n_1(\lambda)} \times \{0,1\}^{n_2(\lambda)} \to \{0,1\}^{n_3(\lambda)}$ の代わりに，以下で定義する $F_{\mathsf{Special}}\colon \{0,1\}^{n_1(\lambda)} \times \{0,1\}^{n_2(\lambda)+1} \to \{0,1\}^{\max\{n_1(\lambda), n_3(\lambda)\}}$ を用いると，構成 3.1 の SDFE は Special Decryption を持つ．

$$F_{\mathsf{Special}}(x,(y,b)) \coloneqq \begin{cases} x & (y \| b = 0^{n_2(\lambda)} \| 0) \\ F(x,y) & (\text{otherwise}) \end{cases}$$

ただし，出力長を揃えるために 0 でパディングするものとする．

証明. Goldwasser らによる Single-Key PKFE の構成（[13], Construction B.1）と同様に成立． □

4. 送受信者編集否認可能公開鍵暗号

4.1 送受信者編集否認可能公開鍵暗号の定義

送信者側は Full モデル，受信者側は Flexible モデルの送受信者編集否認可能公開鍵暗号方式を以下のように定義する．

定義 4.1 (送受信者編集否認可能公開鍵暗号)**.** 平文長 $n(\lambda)$，編集記述長 $\ell(\lambda)$ の PPT 編集関数 $\mathsf{Edit}\colon \{0,1\}^{n(\lambda)} \times \{0,1\}^{\ell(\lambda)} \to \{0,1\}^{n(\lambda)}$ に対する送受信者編集否認可能公開鍵暗号方式 BDEdit は以下の 5 つの PPT アルゴリズム $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{SFake}, \mathsf{RFake})$ からなる．

$\underline{\mathsf{Gen}(1^\lambda) \to (\mathsf{pk}, \mathsf{sk}, \mathsf{dk})}$：セキュリティパラメータ λ を入力にとり，公開鍵 pk，秘密鍵 sk，否認鍵 dk を出力する．

$\underline{\mathsf{Enc}(\mathsf{pk}, m; r) \to \mathsf{ct}}$：公開鍵 pk と平文 $m \in \{0,1\}^{n(\lambda)}$ を入力にとり，暗号文 ct を出力する．

$\underline{\mathsf{Dec}(\mathsf{sk}, \mathsf{ct}) \to m}$：秘密鍵 sk と暗号文 ct を入力にとり，平文 m を出力する．

$\underline{\mathsf{SFake}(\mathsf{pk}, m, r, e) \to r^*}$：公開鍵 pk，元の平文 $m \in \{0,1\}^{n(\lambda)}$，暗号化に使用した乱数 r，編集記述 $e \in \{0,1\}^{\ell(\lambda)}$ を入力にとり，偽の乱数 r^* を出力する．

$\underline{\mathsf{RFake}(\mathsf{dk}, \mathsf{ct}, e) \to \mathsf{sk}_{\mathsf{ct},e}}$：否認鍵 dk，暗号文 ct，編集記述 $e \in \{0,1\}^{\ell(\lambda)}$ を入力にとり，偽の秘密鍵 $\mathsf{sk}_{\mathsf{ct},e}$ を出力する．

送受信者編集否認可能公開鍵暗号方式の正当性，IND-CPA 安全性，送受信者編集否認可能性の定義は以下の通りである．

定義 4.2 (正当性)**.** BDEdit $=$ $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{SFake}, \mathsf{RFake})$ が正当性を満たすとは，任意のセキュリティパラメータ λ，平文 $m \in \{0,1\}^{n(\lambda)}$ に対して以下が成り立つことをいう．

$$\Pr\left[\mathsf{Dec}(\mathsf{sk}, \mathsf{ct}) = m : \begin{matrix}(\mathsf{pk}, \mathsf{sk}, \mathsf{dk}) \leftarrow \mathsf{Gen}(1^\lambda) \\ \mathsf{ct} \leftarrow \mathsf{Enc}(\mathsf{pk}, m; r)\end{matrix}\right] = 1 - \mathsf{negl}(\lambda).$$

定義 4.3 (IND-CPA 安全性)**.** BDEdit $=$ $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{SFake}, \mathsf{RFake})$ が IND-CPA 安全性を満たすとは，任意の

PPT 攻撃者 $\mathcal{A}$ に対して以下が成り立つことをいう．

$$\left|\Pr[\mathsf{CPAGame}_{\mathcal{A}}(\lambda) \to 1] - \frac{1}{2}\right| \leq \mathsf{negl}(\lambda).$$

ただし，$\mathsf{CPAGame}_{\mathcal{A}}(\lambda)$ は PPT 攻撃者 $\mathcal{A}$ と Challenger 間のゲームであって，次のように定義される．

$\mathsf{CPAGame}_{\mathcal{A}}(\lambda)$

Challenger		$\mathcal{A}$
$(\mathsf{pk}, \mathsf{sk}, \mathsf{dk}) \leftarrow \mathsf{Gen}(1^\lambda)$		
	$\xrightarrow{\mathsf{pk}}$	
	$\xleftarrow{m_0, m_1}$	
$b \xleftarrow{\$} \{0,1\}$		
$\mathsf{ct} \leftarrow \mathsf{Enc}(\mathsf{pk}, m_b)$		
	$\xrightarrow{\mathsf{ct}}$	
	$\xleftarrow{b'}$	
$\text{output} \to \begin{cases} 1 & (b' = b) \\ 0 & (b' \neq b) \end{cases}$		

定義 4.4 (送受信者編集否認可能性)．$\mathsf{BDEdit} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{SFake}, \mathsf{RFake})$ が $\delta(\lambda)$-送受信者編集否認可能性を満たすとは，任意の PPT 攻撃者 $\mathcal{A}$ に対して以下が成り立つことをいう．

$$\left|\Pr[\mathsf{BDEditGame}_{\mathcal{A}}(\lambda) \to 1] - \frac{1}{2}\right| \leq \delta(\lambda).$$

ただし，$\mathsf{BDEditGame}_{\mathcal{A}}(\lambda)$ は PPT 攻撃者 $\mathcal{A}$ と Challenger 間のゲームであって，次のように定義される．

$\mathsf{BDEditGame}_{\mathcal{A}}(\lambda)$

Challenger		$\mathcal{A}$
$(\mathsf{pk}, \mathsf{sk}, \mathsf{dk}) \leftarrow \mathsf{Gen}(1^\lambda)$		
	$\xrightarrow{\mathsf{pk}}$	
	$\xleftarrow{m, e}$	
$b \xleftarrow{\$} \{0,1\}$		
$\boldsymbol{X}_0 \coloneqq (\mathsf{Edit}(m,e), r, \mathsf{Enc}(\mathsf{pk}, \mathsf{Edit}(m,e); r), \mathsf{sk})$		
$r^* \leftarrow \mathsf{SFake}(\mathsf{pk}, m, r, \mathsf{Edit}(m,e))$		
$\mathsf{sk}_{\mathsf{ct},e} \leftarrow \mathsf{RFake}(\mathsf{dk}, \mathsf{Enc}(\mathsf{pk}, m; r), e)$		
$\boldsymbol{X}_1 \coloneqq \big(\mathsf{Edit}(m,e), r^*, \mathsf{Enc}(\mathsf{pk}, m; r), \mathsf{sk}_{\mathsf{ct},e}\big)$		
	$\xrightarrow{\boldsymbol{X}_b}$	
	$\xleftarrow{b'}$	
$\text{output} \to \begin{cases} 1 & (b' = b) \\ 0 & (b' \neq b) \end{cases}$		

さらに，送信者または受信者単体に対する編集否認可能性に関して，Goldwasser ら [13] による受信者編集否認可能性の定義を示したうえで，送信者編集否認可能性も新たに定義する．

定義 4.5 (受信者編集否認可能性 [13])．$\mathsf{BDEdit} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{SFake}, \mathsf{RFake})$ が受信者編集否認可能性を満たすとは，任意の PPT 攻撃者 $\mathcal{A}$ に対して以下が成り立つことをいう．

$$\left|\Pr[\mathsf{RDEditGame}_{\mathcal{A}}(\lambda) \to 1] - \frac{1}{2}\right| \leq \mathsf{negl}(\lambda).$$

ただし，$\mathsf{RDEditGame}_{\mathcal{A}}(\lambda)$ は PPT 攻撃者 $\mathcal{A}$ と Challenger 間のゲームであって，次のように定義される．

$\mathsf{RDEditGame}_{\mathcal{A}}(\lambda)$

Challenger		$\mathcal{A}$
$(\mathsf{pk}, \mathsf{sk}, \mathsf{dk}) \leftarrow \mathsf{Gen}(1^\lambda)$		
	$\xrightarrow{\mathsf{pk}}$	
	$\xleftarrow{m, e}$	
$b \xleftarrow{\$} \{0,1\}$		
$\boldsymbol{X}_0 \coloneqq (\mathsf{Enc}(\mathsf{pk}, \mathsf{Edit}(m,e)), \mathsf{sk})$		
$\mathsf{sk}_{\mathsf{ct},e} \leftarrow \mathsf{RFake}(\mathsf{dk}, \mathsf{Enc}(\mathsf{pk}, m), e)$		
$\boldsymbol{X}_1 \coloneqq \big(\mathsf{Enc}(\mathsf{pk}, m), \mathsf{sk}_{\mathsf{ct},e}\big)$		
	$\xrightarrow{\boldsymbol{X}_b}$	
	$\xleftarrow{b'}$	
$\text{output} \to \begin{cases} 1 & (b' = b) \\ 0 & (b' \neq b) \end{cases}$		

定義 4.6 (送信者編集否認可能性)．$\mathsf{BDEdit} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{SFake}, \mathsf{RFake})$ が $\delta(\lambda)$-送信者編集否認可能性を満たすとは，任意の PPT 攻撃者 $\mathcal{A}$ に対して以下が成り立つことをいう．

$$\left|\Pr[\mathsf{SDEditGame}_{\mathcal{A}}(\lambda) \to 1] - \frac{1}{2}\right| \leq \delta(\lambda).$$

ただし，$\mathsf{SDEditGame}_{\mathcal{A}}(\lambda)$ は PPT 攻撃者 $\mathcal{A}$ と Challenger 間のゲームであって，次のように定義される．

$\mathsf{SDEditGame}_{\mathcal{A}}(\lambda)$

Challenger		$\mathcal{A}$
$(\mathsf{pk}, \mathsf{sk}, \mathsf{dk}) \leftarrow \mathsf{Gen}(1^\lambda)$		
	$\xrightarrow{\mathsf{pk}}$	
	$\xleftarrow{m, e}$	
$b \xleftarrow{\$} \{0,1\}$		
$\boldsymbol{X}_0 \coloneqq (\mathsf{Edit}(m,e), r, \mathsf{Enc}(\mathsf{pk}, \mathsf{Edit}(m,e); r))$		
$r^* \leftarrow \mathsf{SFake}(\mathsf{pk}, m, r, \mathsf{Edit}(m,e))$		
$\boldsymbol{X}_1 \coloneqq \big(\mathsf{Edit}(m,e), r^*, \mathsf{Enc}(\mathsf{pk}, m; r)\big)$		
	$\xrightarrow{\boldsymbol{X}_b}$	
	$\xleftarrow{b'}$	
$\text{output} \to \begin{cases} 1 & (b' = b) \\ 0 & (b' \neq b) \end{cases}$		

4.2 提案構成

Goldwasser らによる受信者編集否認可能公開鍵暗号の構成（[13], Construction 4.2）をもとに，送受信者編集否認可能公開鍵暗号の構成を提案する．なお，構成には，$F = F_{\mathsf{Edit}}$ とした $F_{\mathsf{Special}} \colon \{0,1\}^{n(\lambda)+(\lambda+\ell(\lambda))} \times \{0,1\}^{\lambda+\ell(\lambda)+1} \to \{0,1\}^{n(\lambda)+(\lambda+\ell(\lambda))}$（定理 3.4）に対する送信者否認可能

Single-Key PKFE 方式 $\mathsf{SDFE} = (\mathsf{Setup}, \mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{SFake})$ を用いる．

構成 4.1. $\underline{\mathsf{BDEdit.Gen}(1^\lambda) \to (\mathsf{pk}, \mathsf{sk}, \mathsf{dk})}$：セキュリティパラメータ λ を入力にとり，以下を実行する．

1. $(\mathsf{mpk}, \mathsf{msk}) \leftarrow \mathsf{SDFE.Setup}(1^\lambda)$.
2. $y \xleftarrow{\$} \{0,1\}^{\lambda+\ell(\lambda)}$.
3. $\mathsf{sk}_y \leftarrow \mathsf{SDFE.Gen}(\mathsf{msk}, y)$.
4. $(\mathsf{pk} := \mathsf{mpk}, \mathsf{sk} := \mathsf{sk}_y, \mathsf{dk} := \mathsf{msk})$ を出力．

$\underline{\mathsf{BDEdit.Enc}(\mathsf{pk}, m; r) \to \mathsf{ct}}$：公開鍵 pk と平文 $m \in \{0,1\}^{n(\lambda)}$ を入力にとり，以下を実行する．

1. $k \xleftarrow{\$} \{0,1\}^{\lambda+\ell(\lambda)}$.
2. $\mathsf{ct} \leftarrow \mathsf{SDFE.Enc}(\mathsf{mpk}, (m,k); r_{\mathsf{SDFE}})$ を出力．

$\underline{\mathsf{BDEdit.Dec}(\mathsf{sk}, \mathsf{ct}) \to m}$：秘密鍵 sk と暗号文 ct を入力にとり，以下を実行する．

1. $m = \mathsf{SDFE.Dec}(\mathsf{sk}_y, \mathsf{ct})$ を出力

$\underline{\mathsf{BDEdit.SFake}(\mathsf{pk}, m, r, e) \to r^*}$：公開鍵 pk，元の平文 $m \in \{0,1\}^{n(\lambda)}$，暗号化に使用した乱数 $r = (k, r_{\mathsf{SDFE}})$，編集記述 $e \in \{0,1\}^{\ell(\lambda)}$ を入力にとり，以下を実行する．

1. $r^*_{\mathsf{SDFE}} \leftarrow \mathsf{SDFE.SFake}(\mathsf{mpk}, m, r_{\mathsf{SDFE}}, \mathsf{Edit}(m,e))$.
2. $r^* = (k, r^*_{\mathsf{SDFE}})$ を出力．

$\underline{\mathsf{BDEdit.RFake}(\mathsf{dk}, \mathsf{ct}, e) \to \mathsf{sk}_{\mathsf{ct},e}}$：否認鍵 dk，暗号文 ct，編集記述 $e \in \{0,1\}^{\ell(\lambda)}$ を入力にとり，以下を実行する．

1. $(m,k) = \mathsf{SDFE.Dec}(\mathsf{msk}, \mathsf{ct})$.
2. $y := k \oplus (0^\lambda \| e)$.
3. $\mathsf{sk}_y \leftarrow \mathsf{SDFE.Gen}(\mathsf{msk}, y)$.
4. $\mathsf{sk}_{\mathsf{ct},e} := \mathsf{sk}_y$ を出力．

定理 4.1. 構成 4.1 の BDEdit は正当性を満たす．

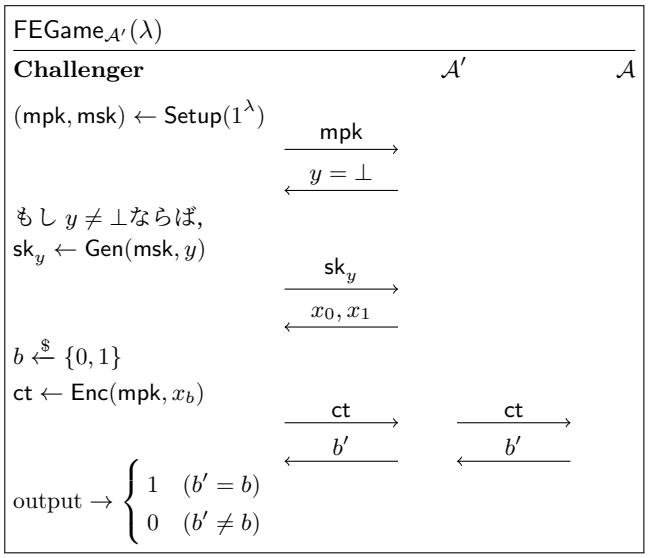
証明. 任意の $\lambda, m \in \{0,1\}^{n(\lambda)}$ に対して，

$$
\begin{aligned}
&\Pr\left[\begin{array}{l} m = \mathsf{BDEdit.Dec}(\mathsf{sk}, \mathsf{ct}) : \\ \quad (\mathsf{pk}, \mathsf{sk}, \mathsf{dk}) \leftarrow \mathsf{BDEdit.Gen}(1^\lambda) \\ \quad \mathsf{ct} \leftarrow \mathsf{BDEdit.Enc}(\mathsf{pk}, m) \end{array}\right] \\
&= \Pr\left[\begin{array}{l} m = \mathsf{SDFE.Dec}(\mathsf{sk}, \mathsf{ct}) : \\ \quad (k,y) \xleftarrow{\$} \{0,1\}^{\lambda+\ell(\lambda)} \times \{0,1\}^{\lambda+\ell(\lambda)} \\ \quad (\mathsf{mpk}, \mathsf{msk}) \leftarrow \mathsf{SDFE.Setup}(1^\lambda) \\ \quad \mathsf{sk}_y \leftarrow \mathsf{SDFE.Gen}(\mathsf{msk}, y) \\ \quad \mathsf{ct} \leftarrow \mathsf{SDFE.Enc}(\mathsf{mpk}, (m,k)) \end{array}\right] \\
&= \Pr\left[\begin{array}{l} m = F_{\mathsf{Edit}}((m,k), y) : \\ \quad (k,y) \xleftarrow{\$} \{0,1\}^{\lambda+\ell(\lambda)} \times \{0,1\}^{\lambda+\ell(\lambda)} \end{array}\right] \\
&= 1 - \Pr\left[\begin{array}{l} \mathsf{Edit}(m,e) = F_{\mathsf{Edit}}((m,k), y) : \\ \quad (k,y) \xleftarrow{\$} \{0,1\}^{\lambda+\ell(\lambda)} \times \{0,1\}^{\lambda+\ell(\lambda)} \end{array}\right] \\
&= 1 - \Pr\left[\begin{array}{l} \exists e \text{ s.t. } y \oplus k = 0^\lambda \| e : \\ \quad (k,y) \xleftarrow{\$} \{0,1\}^{\lambda+\ell(\lambda)} \times \{0,1\}^{\lambda+\ell(\lambda)} \end{array}\right] \\
&= 1 - \frac{2^\lambda}{2^\lambda \cdot 2^\lambda} = 1 - \frac{1}{2^\lambda} = 1 - \mathsf{negl}(\lambda).
\end{aligned}
$$

□

定理 4.2. 構成 4.1 の BDEdit は IND-CPA 安全性を満たす．

証明. IND-CPA 安全性を破る PPT 攻撃者 $\mathcal{A}$ を用いて，送信者否認可能 Single-Key PKFE の安全性を破る攻撃者 $\mathcal{A}'$ を構成することで，定理を示す．$\mathsf{FEGame}_{\mathcal{A}'}(\lambda)$ において，攻撃者 $\mathcal{A}'$ が $y = \bot$ を選択すると，IND-CPA 安全のゲーム $\mathsf{CPAGame}_{\mathcal{A}}(\lambda)$ に一致する．よって，攻撃者 $\mathcal{A}'$ は Challenger から送られてきた ct をそのまま IND-CPA 安全性を破る PPT 攻撃者 $\mathcal{A}$ にクエリすることで，送信者否認可能 Single-Key PKFE の安全性を破ることができる．



□

定理 4.3. SDFE が $\delta(\lambda)$-送信者否認可能であるとき，構成 4.1 の BDEdit は $\delta(\lambda)$-送信者編集否認可能である．

証明. BDEdit.SFake で出力される偽の乱数 $r^* = (k, r^*_{\mathsf{SDFE}})$ のうち，BDEdit の送信者否認可能性に関わるのは r^*_{SDFE} のみであり，r^*_{SDFE} は SDFE.SFake によって生成される．よって，SDFE が $\delta(\lambda)$-送信者否認可能であるならば，構成 4.1 の BDEdit は $\delta(\lambda)$-送信者否認可能である． □

定理 4.4. 構成 4.1 の BDEdit は受信者編集否認可能である．

証明. [13] の Theorem 4.3 より成立． □

定理 4.5. SDFE が $\delta(\lambda)$-送信者否認可能であるとき，構成 4.1 の BDEdit は $\delta(\lambda)$-送受信者編集否認可能である．

証明. 定理 4.3 と定理 4.4 より成立． □

5. まとめと今後の課題

本研究では Goldwasser ら [13] による受信者編集否認可能公開鍵暗号を拡張した送受信者編集否認可能公開鍵暗号を提案し, 公開鍵暗号, 送信者否認可能公開鍵暗号, Garbled Circuit による具体的構成法も併せて提案した.

今後の課題として, 効率の改善や, より強い否認可能性を満たす方式の検討が挙げられる. 効率面では, Goldwasser ら [13] の構成, 及び本研究の構成では Garbled Ciricuit を使用しているため, 暗号文サイズが大きくなってしまっている. Garbled Circuit をブラックボックスで使用しているため, 近年の Free XOR [15] や Half Gates [22] に代表される, Garbled Circuit の効率改善を目的とした研究 [11], [16], [19] と組み合わせることで効率の改善が見込める. また, より強い否認可能性について, 送受信者否認可能暗号に関する先行研究 [7], [8], [18] で指摘されているのと同様に, 本研究で提案した送受信者編集否認可能性でも, 送信者と受信者が同じ平文を主張する場合のみを捉えている. 送信者と受信者が異なる平文を主張する場合の否認可能性を考えるのが Off-the-record 否認可能性 [7] である. この Off-the-record 否認可能性は, 攻撃者が得る通信記録が送受信者のどちらが真実を述べているのかを判断するのに役に立たないことを保証する. Off-the-record 否認可能性を満たす送受信者編集否認可能公開鍵暗号の構成法も, 今後の研究課題の 1 つである.

謝辞 本研究は JSPS 科研費 JP23H00468, JP23H00479, JP23K17455, JP23K21644, JP23K24846 の助成, および 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] Agrawal, S., Goldwasser, S. and Mossel, S.: Deniable fully homomorphic encryption from learning with errors, *Annual International Cryptology Conference*, Springer, pp. 641–670 (2021).

[2] An, Z. and Zhang, F.: Deny whatever you want: dual-deniable public-key encryption, *IACR International Conference on Public-Key Cryptography*, Springer, pp. 246–279 (2025).

[3] Apon, D., Fan, X. and Liu, F.-H.: Deniable attribute based encryption for branching programs from LWE, *Theory of Cryptography Conference*, Springer, pp. 299–329 (2016).

[4] Bellare, M., Hoang, V. T. and Rogaway, P.: Foundations of garbled circuits, *Proceedings of the 2012 ACM conference on Computer and communications security*, pp. 784–796 (2012).

[5] Bendlin, R., Nielsen, J. B., Nordholt, P. S. and Orlandi, C.: Lower and upper bounds for deniable public-key encryption, *International Conference on the Theory and Application of Cryptology and Information Security*, Springer, pp. 125–142 (2011).

[6] Boneh, D., Sahai, A. and Waters, B.: Functional encryption: Definitions and challenges, *Theory of Cryptography Conference*, Springer, pp. 253–273 (2011).

[7] Canetti, R., Park, S. and Poburinnaya, O.: Fully deniable interactive encryption, *Annual International Cryptology Conference*, Springer, pp. 807–835 (2020).

[8] Canetti, R., Dwork, C., Naor, M. and Ostrovsky, R.: Deniable encryption, *Annual International Cryptology Conference*, Springer, pp. 90–104 (1997).

[9] Dachman-Soled, D.: On minimal assumptions for sender-deniable public key encryption, *International Workshop on Public Key Cryptography*, Springer, pp. 574–591 (2014).

[10] De Caro, A., Iovino, V. and O'Neill, A.: Deniable functional encryption, *Public-Key Cryptography–PKC 2016: 19th IACR International Conference on Practice and Theory in Public-Key Cryptography, Taipei, Taiwan, March 6-9, 2016, Proceedings, Part I*, Springer, pp. 196–222 (2016).

[11] Dietz, M., Li, H. and Lin, H.: TinyLabels: How to Compress Garbled Circuit Input Labels, Efficiently, *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, Springer, pp. 245–274 (2025).

[12] Goldwasser, S., Kalai, Y., Popa, R. A., Vaikuntanathan, V. and Zeldovich, N.: Reusable garbled circuits and succinct functional encryption, *Proceedings of the forty-fifth annual ACM symposium on Theory of computing*, pp. 555–564 (2013).

[13] Goldwasser, S., Klein, S. and Wichs, D.: The edited truth, *Theory of Cryptography Conference*, Springer, pp. 305–340 (2017).

[14] Gorbunov, S., Vaikuntanathan, V. and Wee, H.: Functional encryption with bounded collusions via multi-party computation, *Annual Cryptology Conference*, Springer, pp. 162–179 (2012).

[15] Kolesnikov, V. and Schneider, T.: Improved garbled circuit: Free XOR gates and applications, *International Colloquium on Automata, Languages, and Programming*, Springer, pp. 486–498 (2008).

[16] Liu, H., Wang, X., Yang, K. and Yu, Y.: BitGC: garbled circuits with 1 bit per gate, *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, Springer, pp. 437–466 (2025).

[17] O'Neill, A.: Definitional issues in functional encryption, *Cryptology ePrint Archive* (2010).

[18] O'Neill, A., Peikert, C. and Waters, B.: Bi-deniable public-key encryption, *Annual Cryptology Conference*, Springer, pp. 525–542 (2011).

[19] Rosulek, M. and Roy, L.: Three halves make a whole? Beating the half-gates lower bound for garbled circuits, *Annual International Cryptology Conference*, Springer, pp. 94–124 (2021).

[20] Sahai, A. and Seyalioglu, H.: Worry-free encryption: functional encryption with public keys, *Proceedings of the 17th ACM conference on Computer and communications security*, pp. 463–472 (2010).

[21] Yao, A. C.-C.: How to generate and exchange secrets, *27th annual symposium on foundations of computer science (Sfcs 1986)*, IEEE, pp. 162–167 (1986).

[22] Zahur, S., Rosulek, M. and Evans, D.: Two halves make a whole: Reducing data transfer in garbled circuits using half gates, *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, Springer, pp. 220–250 (2015).