マルチパーティ公平交換プロトコルの安全性解析と改善

藤阪 祐飛[1,a)]　中井 雄士[1]　鈴木 幸太郎[1]

概要：公平交換とは，どちらか一方による持ち逃げができないことを保証して，2 人の参加者それぞれが持つ電子的なアイテムの交換を可能とする暗号プロトコルである．マルチパーティ公平交換プロトコルとは，公平交換における参加者数を一般の n 人に拡張したものである．Kılınç と Küpçü (TOPS 2021) は，信頼できる第三者 (TTP) が存在することを仮定する Optimistic モデルの基で，任意の交換トポロジーに適用可能で効率的なマルチパーティ公平交換プロトコルを提案した．本研究では，Kılınç らの方式において，2 人以上の参加者を corrupt した攻撃者が正直な参加者のアイテムを持ち逃げできることを指摘する．攻撃手法は，TTP の処理内容が事前に設定された制限時間毎に変化する一方で，問題が発生しない限り参加者らはその時間に依らず処理を継続してよいとした Kılınç らの方式の性質を利用する．さらに，本研究では指摘した問題を修正する手法を 2 つ提案する．一方は制限時間に基づく待機を要求する手法であり，もう一方は Kılınç らの方式よりも通信量が増加する代わりに制限時間に基づく待機を要求しない手法である．

キーワード：マルチパーティ公平交換，Optimistic モデル，エスクロー

Security Analysis and Improvement of a Multi-party Fair Exchange Protocol

Yuhi Fujisaka[1,a)]　Takeshi Nakai[1]　Koutarou Suzuki[1]

Abstract: Fair exchange enables two mutually distrustful parties to exchange electronic items in a fair manner. Multi-party fair exchange is a generalization of fair exchange regarding the number of parties. Kılınç and Küpçü (TOPS 2021) proposed an efficient multi-party fair exchange protocol applicable to any exchange topology based on the optimistic model, assuming the existence of a trusted third party (TTP). In this work, we highlight that in their protocol, an adversary who corrupts two or more parties can steal an honest party's item without revealing its own. Our attack exploits the feature where the TTP's process depends on predetermined time intervals, while parties can continue their procedures regardless of the time as long as no issues arise. Furthermore, we propose two methods to address the issue. One method requires parties to wait for the time intervals. The other one does not require such waiting, instead of increasing the communication cost.

Keywords: Multi-party fair exchange, Optimistic model, Escrow

1. はじめに

1.1 背景

公平交換プロトコルとは，2 人の参加者がそれぞれ持つ電子データ (以降，アイテムと呼ぶ) を公平に交換することを目的とする暗号プロトコルである．このプロトコルは，どちらか一方の参加者のみが相手の電子データを得てプロトコルを終了する攻撃 (以降，持ち逃げ攻撃と呼ぶ) が不可能であることをその安全性として保証する [1, 2]．公平交換は，信頼できる第三者 (Trusted Third Party: TTP) などの仮定がなければ実現不可能であることが知られている [3]．この不可能性を回避するための代表的な手法とし

[1] 豊橋技術科学大学，〒 441-8580 愛知県豊橋市天伯町雲雀ヶ丘 1-1
Toyohashi University of Technology, 1-1 Hibarigaoka, Tempakucho, Toyohashi-shi, Aichi, 441-8580, Japan.
[a)] fujisaka.yuhi.dp@tut.jp

て，Optimistic モデルがある [1, 4]．Optimistic モデルとは，プロトコルの参加者以外に TTP の存在を仮定するものである．この TTP は公平性を満たすための補助的な役割を担い，すべての参加者が正直な場合はプロトコルへ介入せず，問題が検出された場合のみ不正な参加者による持ち逃げ攻撃を防ぐためにプロトコルへ介入する．

マルチパーティ公平交換プロトコル (Multiparty Fair Exchange: MFE) は公平交換における参加者数を一般の $n \geq 2$ に拡張したものであり，Asokan–Schunter–Waidner [5] によって初めて提案された．なお，Asokan らの方式は Optimistic モデルに基づく．交換の様子が参加者らをノードとした有向グラフ (以降，交換トポロジーと呼ぶ) で表されるものとしたとき，Asokan らの方式は任意の交換トポロジーに適用可能である．その後，交換トポロジーをリングに制限して効率化された MFE プロトコルが提案された [6–8]．

Kılınç と Küpçü [9] は，任意の交換トポロジーに適用可能で，Asokan らの方式 [5] よりも効率的な，Optimistic モデルに基づく MFE プロトコル (以降，KK-MFE と呼ぶ) を提案した．KK-MFE のラウンド数と通信コストは，n に関する漸近評価で最適である．さらに，Asokan らの方式はブロードキャストチャネルを要求する一方で，KK-MFE はこれを必要としない利点を持つ．我々が知る限りでは，任意の交換トポロジーに適用可能な MFE プロトコルは，Asokan らの方式と KK-MFE のみである．

1.2 本研究の貢献

本研究では，KK-MFE において，2 人以上の不正な参加者が結託すると持ち逃げ攻撃が可能であることを示す．つまり，KK-MFE は MFE の安全性を満たさないことを示す．KK-MFE では，TTP の処理内容は事前に設定された制限時間に依存して変化する一方，参加者らは問題が発生しない限りその時間に依らず手順を継続してよい（つまり設定時間に基づく待機を要求しない）としている．我々が示す攻撃手法では，この性質を悪用することで，公平性を侵害するふるまいが検出されたとしても，正直な参加者らが TTP を呼び出せず持ち逃げ攻撃が成功する状況が発生し得ることを示す．

さらに，提案の攻撃手法への対策を 2 つ提案する．一つは，攻撃手法が KK-MFE では参加者らの手順が TTP の設定時間に依存しないことを悪用していることに着目し，これへの対策として参加者らに制限時間に基づく待機を要求する手法である．なお，TTP の設定時間は，不正なふるまいに対する処理を TTP が受け付ける制限時間を表しており，そのため，十分に余裕をもった時間設定にすることが望ましい．しかし，この提案手法ではプロトコルの実行がその設定時間に直接的に依存することとなり，それに伴い実行時間の効率が悪くなる欠点がある．したがって，もう一つの手法として，KK-MFE よりも通信量が増加する代わりに TTP の設定時間に基づく待機を要求しない手法を提案する．提案手法はそれぞれ，MFE の安全性定義を満たすことが期待されるが，形式的な安全性証明については今後の研究課題とする．

1.3 本稿の構成

2 節以降の本稿の構成は以下の通りである．2 節では，本稿で用いる記法，安全性定義および暗号プリミティブを導入する．3 節では，KK-MFE の概要を説明する．特に簡単のため 3 者間の場合を例示し，本研究の貢献の一つである攻撃手法で利用するポイントを示す．ここで示したポイントを基に，4 節で KK-MFE に対する攻撃手法を示す．さらに，5 節では，その攻撃に対する対策手法を 2 つ提案する．最後に 6 節はまとめである．

2. 準備

2.1 記法と設定

正整数 i について，$[i] := \{1, \ldots, i\}$ とする．行列 Υ に対し，$\Upsilon[a, b]$ で Υ の a 行 b 列の値を示すとする．セキュリティパラメータを l で表し，すべての参加者は l に関する確率的多項式時間 (Probabilistic Polynomial Time: PPT) アルゴリズムであるとする．本稿で扱うプロトコルでは，任意の参加者間はセキュア認証チャネルで接続されているものとする．なお，ブロードキャストチャネルの存在は仮定しない．プロトコルの参加者数を n で表す．corrupt された (あるいは不正な) 参加者の数を c としたとき，$c < n$ とする．時刻を正整数 t で表し，すべての参加者は共通の現在時刻 t_{now} を参照できるものと仮定する．

2.2 検証可能な (ラベル付き) 公開鍵暗号

検証可能な (ラベル付き) 公開鍵暗号 VPKE は，4 つの PPT アルゴリズム $(\mathsf{PkGen}, \mathsf{PkPrvEnc}, \mathsf{PkVrfy}, \mathsf{PkDec})$ の組で定義される．

$\mathsf{PkGen}(1^l) \to (\mathsf{SK}_\mathsf{T}, \mathsf{PK}_\mathsf{T})$: セキュリティパラメータ l を入力として，秘密鍵と公開鍵のペア $(\mathsf{SK}_\mathsf{T}, \mathsf{PK}_\mathsf{T})$ を出力する．

$\mathsf{PkPrvEnc}(\mathsf{PK}_\mathsf{T}, w; \mathrm{lbl}) \to (\mathsf{VS}, \mathsf{VSproof})$: 公開鍵 PK_T，平文 w，述語 ψ，ラベル lbl を入力として，暗号文 VS とその検証 VSproof を出力する．

$\mathsf{PkVrfy}(\mathsf{PK}_\mathsf{T}, \psi, \mathsf{VS}, \mathsf{VSproof}) \to \mathrm{T/F}$: $\mathsf{PK}_\mathsf{T}, \mathsf{VS}, \mathsf{VSproof}$, ψ を入力として T または F を出力する．

$\mathrm{PkDec}(\mathsf{SK}_\mathsf{T}, \mathsf{VS}) \to (w, \mathrm{lbl}) = \mathrm{T}$: SK_T と VS を入力として，w と lbl を出力する．

VPKE は，復号の正当性 (正当な鍵ペアであれば常に復号に成功すること) と秘匿性 (暗号文から平文の情報が漏洩しないこと) に加えて以下の性質を満たすものとする (詳細な定義は [9, 10] を参照)．

$$\mathsf{PkVrfy}(\mathsf{PK}_\mathsf{T}, \psi, \mathsf{VS}, \mathsf{VSproof}) \to \mathrm{T} \iff$$
$$\mathsf{PkPrvEnc}(\mathsf{PK}_\mathsf{T}, w; \mathrm{lbl}) \to (\mathsf{VS}, \mathsf{VSproof}) \wedge \psi(w) = 1.$$

2.3 検証可能な (n, n) 閾値暗号

検証可能な (n,n)-閾値暗号 TPKE は，7 つの PPT アルゴリズム $(\mathsf{ThGen}, \mathsf{ThPrvEnc}, \mathsf{ThVrfy}, \mathsf{ThDShr}, \mathsf{ThDSPrv}, \mathsf{ThDSVrfy}, \mathsf{ThDec})$ の組で定義される．

$\mathsf{ThGen}(1^l, n, n) \to (\mathsf{pp}, v, \{x_i\}_{1\le i\le n})$: セキュリティパラメータ l と閾値 n を入力として，公開パラメータ pp，検証鍵 v，n 個の秘密鍵 $\{x_i\}_{1\le i\le n}$ を出力する．

$\mathsf{ThPrvEnc}(\mathsf{pp}, m, \gamma) \to (\mathsf{VE}, \mathsf{VEproof})$: 公開パラメータ pp, 平文 m, 述語 γ を入力として, 暗号文 VE とその証明 VEproof を出力する．

$\mathsf{ThVrfy}(\mathsf{pp}, \gamma, \mathsf{VE}, \mathsf{VEproof}) \to \mathrm{T/F}$: pp，述語 γ，VE, VEproof を入力として T または F を出力する．

$\mathsf{ThDShr}(x_i, \mathsf{pp}, \mathsf{VE}) \to (d_i, \mathsf{DSproof}_i)$: x_i, pp, VE を入力として，復号鍵のシェア d_i と証明 $\mathsf{DSproof}_i$ を出力する．なお，シェアの閾値は n である．

$\mathsf{ThDSVrfy}(v, i, \mathsf{pp}, \mathsf{VE}, d_i, \mathsf{DSproof}_i) \to \mathrm{T/F}$: v, pp, VE, d_i, $\mathsf{DSproof}_i$ を入力として T または F を出力する．

$\mathsf{ThDec}(\{d_i\}_{1\le i\le n}, \mathsf{pp}, \mathsf{VE}) \to m$: n 個の復号鍵のシェア $\{d_i\}_{1\le i\le n}$, pp, VE を入力として，平文 m を出力する．

TPKE は，復号の正当性 (正当な n 個の復号鍵のシェアを用いれば常に復号に成功すること) と秘匿性 ($n-1$ 個以下の秘密鍵と暗号文から平文の情報が漏洩しないこと) に加えて以下の性質を満たすものとする (詳細な定義は [9,11] を参照)．

- $\mathsf{ThVrfy}(\mathsf{pp}, \gamma, \mathsf{VE}, \mathsf{VEproof}) \to \mathrm{T} \iff$ $\mathsf{ThPrvEnc}(\mathsf{pp}, m, \gamma) \to (\mathsf{VE}, \mathsf{VEproof}) \wedge \gamma(m) = 1$
- $\mathsf{ThDSVrfy}(v, i, \mathsf{pp}, \mathsf{VE}, d_i, \mathsf{DSproof}_i) \to \mathrm{T} \iff$ $\mathsf{ThDShr}(x_i, \mathsf{pp}, \mathsf{VE}) \to (d_i, \mathsf{DSproof}_i)$

なお，本稿では述語 γ はアイテムの正当性を確認するための検証式に対応する．

2.4 MFE の安全性

本稿では簡単のため，参加者 P_i が送信するアイテムは高々 1 種類 (f_i とする) であるとする．つまり，P_1 が P_2, P_3 それぞれにアイテムを送信するとしても，送るアイテムは同一であるとする．プロトコルで扱うアイテムの識別子の全体集合を $I \subseteq [n]$ とする．MFE における交換の目標を有向グラフの隣接行列 (交換トポロジー) として表現する．交

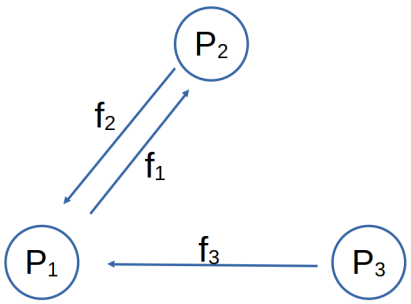

図 1 交換トポロジーの例: 参加者 P_i がアイテム f_i を初期値として持つ．P_1 が f_2 と f_3 を受け取り，P_2 が f_1 を受け取ることを目的とする．なお，P_3 は何も受け取らない．

換トポロジー Υ は $n \times n$ 行列であり，行と列はそれぞれ参加者に割り当てられ，各セルの値は 0 または 1 である．$\Upsilon[i,j] = 1$ であるとき，参加者 P_i は P_j へ自身のアイテムを送信することを示す．例えば，図 1 の交換トポロジーは以下の行列で表される．

$$\Upsilon = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix}.$$

MFE の安全性は，(公平性[*1]を保証する) 秘密計算プロトコルの安全性の定式化と同様に，Real/Ideal パラダイムによって定式化される．Real 世界は安全性証明の対象となるプロトコル π とそれに対する攻撃者 $\mathcal{A}$ との間の試行を表し，Ideal 世界は π が実現したい理想機能 $\mathcal{F}$ とシミュレータ $\mathcal{S}$ との間の試行を表す．このとき，直感的には，π が安全であるとは，任意の攻撃者 $\mathcal{A}$ に対し，$\mathcal{A}$ と同様の出力を得られるシミュレータ $\mathcal{S}$ が存在することを意味する．

MFE の理想機能はアルゴリズム 1 で定義される．正直な参加者らの集合を H，不正な参加者らの集合を C とする．$\phi = (\phi_1, \ldots, \phi_n)$ は，交換トポロジーに基づいて定義される関数の組である．各 ϕ_i は交換で扱うすべてのアイテムを入力値として取り，その出力で参加者 P_i が受け取るべきアイテムを表す．図 1 の例に基づけば，$\phi_1(f_1, f_2, f_3) = f_1 || f_3$, $\phi_2(f_1, f_2, f_3) = f_2$，$\phi_3(f_1, f_2, f_3) = \perp$ である．

直感的には，MFE プロトコルが安全であるとは，その終了時点で必ず以下の 2 つのどちらか一方を満たすことをいう．

- すべての参加者が望むアイテムを得られる．
- すべての参加者が望むアイテムを得られない．

つまり，一部の交換のみが実行された場合は安全とみなさない．例えば，図 1 の例では，P_1 と P_2 間の交換が公平に

[*1] 秘密計算における公平性とは，すべての参加者が出力を得るか，すべての参加者が出力を得ないかのどちらか一方しか起き得ないことを保証する安全性要件である．

なされたとしても，P_1 がアイテム f_3 を得られなかった場合は安全とはみなさない．

アルゴリズム 1 MFE の理想機能 $\mathcal{F}_{\mathrm{mfe}}^{\phi}$

(1) 以下すべてのメッセージを受け取るまで待機する．
- $i \in H$ について，正直な参加者 P_i からアイテム f_i
- $j \in C$ について，シミュレータ $\mathcal{S}$ からアイテム f_j またはメッセージ abort

(2) もしシミュレータからの入力に abort が含まれる場合は，$\perp$ をすべての参加者に送信しプロセスを終了する．

(3) それ以外の場合，$i \in [n]$ について，参加者 P_i に $\phi_i(f_1, \ldots, f_n)$ を出力する．

3. 既存方式 (KK-MFE) の概要

3.1 方式の概要

KK-MFE は，(1) セットアップ，(2) 暗号化アイテム交換，(3) 復号鍵シェアエスクロー交換，(4) 復号鍵シェア交換 の 4 フェーズからなる．以下にそれぞれのフェーズの概要を示す．

(1) セットアップ: TTP は $\mathsf{PkGen}(1^l) \to (\mathsf{SK_T}, \mathsf{PK_T})$ を実行し $\mathsf{PK_T}$ を公開する．n 人の参加者らで $\mathsf{ThGen}(1^l, n, n) \to (\mathsf{pp}, v, \{x_i\}_{1 \le i \le n})$ を実行し，参加者 P_i が x_i を受け取り，(pp, v) を公開値とする．

(2) 暗号化アイテム交換: 各参加者は自身が保持するアイテムを ThPrvEnc で暗号化し，その暗号文を**すべての**参加者へ送信する．アイテム f_i の暗号文を VE_i とする．

(3) 復号鍵シェアエスクロー交換: すべての $i \in I$ について，各参加者 P_k は ThDShr で VE_i の復号鍵のシェア d_k^i を生成し，さらに $\mathsf{PkPrvEnc}_{\mathsf{PK_T}}$ で d_k^i を暗号化した暗号文 VS_k^i(エスクローと呼ぶ) を生成する．その後，VS_k^i を**対応するアイテムを要求する参加者** (つまり，$\Upsilon[i,j] = 1$ である P_j) へ送信する．図 1 の P_3 のようにアイテムを受け取らない参加者は，ここでは他参加者からシェアを受け取らない．

(4) 復号鍵シェア交換: 各参加者は，(3) で生成した復号鍵のシェアを対応するアイテムを要求する参加者へ送信する．他参加者から受け取った復号鍵のシェアを用いて，(2) で受け取った暗号文を復号しアイテムを得る．

上記の手順はすべての参加者が手順通りに動作した場合を前提としている．したがって，セットアップ以降 TTP はプロトコルに介入しない．フェーズ (3) と (4) で問題が生じた場合は，参加者らは TTP を呼び出して問題の解決を図る．以下に TTP の役割の概要を示す．なお，t_1 と t_2 は事前に設定された時間のパラメータである $(t_1 < t_2)$．

アルゴリズム 2 Resolve 1

Input: $(\mathsf{Resolve\,1}, \mathrm{lbl}, \{u, \mathsf{VE}_u\}_{u \subseteq I}, j \in [n])$ from P_i
1: Parse lbl := $id||\Upsilon||t_1||t_2||\mathsf{pp}||v$
2: **if** $t_{now} < t_1$ **or** $\forall u, \mathsf{ThVrfy}(\mathsf{pp}, \gamma_u, \mathsf{VE}_u, \mathsf{VEproof}_u) \to \mathrm{T}$ **then**
3: **if** DB[lbl] = empty **then**
4: Initialize complaintList and ε as empty lists.
5: **for all** $(u, j) \in I \times [n]$ **do**
6: DS$[u, j] \leftarrow 1$
7: DB[lbl]:= (complaintList, ε, DS)
8: **Refer** DB[lbl] = (complaintList, ε, DS)
9: $\varepsilon[u] := \mathsf{VE}_u$ **for all** u s.t.$\Upsilon[u, i] = 1$
10: DS$[u, j] :=\perp$ **for all** u s.t.$\Upsilon[u, i] = 1$
11: complaintList := complaintList $\cup \{(P_i, P_j)\}$
12: **return** “Resolve2 を実行”
13: **else**
14: **return** abort

アルゴリズム 3 Resolve 2

Input: $(\mathsf{Resolve\,2}, \mathrm{lbl}, \mathcal{V} = (u, j, \mathsf{VS}_u^j, \mathsf{VSproof}_u^j)_{u,j \subseteq I})$ from P_i
1: Parse lbl := $id||\Upsilon||t_1||t_2||\mathsf{pp}||v$
2: **if** $t_1 < t_{now} < t_2$ **then**
3: **if** $\exists u \in I, \mathrm{DS}[u, j] =\perp$ **then**
4: $(d_u^j, \mathsf{DSproof}_u^j) \leftarrow \mathsf{PkDec}(\mathsf{SK_T}, \mathsf{VS}_u^j)$
5: $\mathsf{VE}_u \leftarrow \varepsilon[u]$
6: **if** $\mathsf{ThDSVrfy}(v, \mathsf{pp}, \mathsf{VE}_u, d_u^j, \mathsf{DSproof}_u^j) \to \mathrm{T}$ **then**
7: DS$[u, j] := (d_u^j, \mathsf{DSproof}_u^j)$
8: **else**
9: **return** “検証に失敗”

アルゴリズム 4 Resolve 3

Input: $(\mathsf{Resolve\,3}, \mathrm{lbl}, \mathcal{V} = (\{u, \mathsf{VS}_u^j, \mathsf{VSproof}_u^j)_{u \subseteq I}$ **or** $0\}, j \in [n])$ from P_i
1: Parse lbl := $id||\Upsilon||t_1||t_2||\mathsf{pp}||v$
2: **if** $\forall (x, y) \in I \times [n], \mathrm{DS}[x, y] \neq\perp$ **and** $t_1 < t_{now}$ **then**
3: **if** $\mathcal{V} = 0$ **then**
4: $(d_u^j, \mathsf{DSproof}_u^j) \leftarrow \mathrm{DS}[u, j]$ **for all** u s.t. $(P_i, P_j) \in$ complaintList **and** $\Upsilon[u, i] = 1$
5: **return** $(d_u^i, \mathsf{DSproof}_u^i)$
6: **else**
7: **if** $\mathsf{PkVrfy}(\mathsf{PK_T}, \gamma_u, \mathsf{VS}_u^j, \mathsf{VSproof}_u^j) \to \mathrm{T}$ **then**
8: $(d_u^j, \mathsf{DSproof}_u^j) \leftarrow \mathsf{PkDec}(\mathsf{SK_T}, \mathsf{VS}_u^j)$
9: **return** $(d_u^j, \mathsf{DSproof}_u^j)$
10: **else**
11: **return** “検証に失敗”
12: **else if** $t_{now} > t_2$ **then**
13: **return** abort

TTP の役割: フェーズ (3) において P_i からエスクローを受け取れなかった参加者 P_j がいた場合，P_j はその事実を TTP に報告する．TTP は (i, j) で問題が起きたことを内部のリストに記録し，さらに P_i へ報告の内容を通達する．なお，TTP はこの処理を時刻 t_1 まで受け付ける．不正を告発された P_i は，時刻 t_1 から t_2 の間に，TTP へ対応する (本来 P_j へ送るべきであった) エスクローを TTP へ送ることでこの問題を解消し，リストからその記録を消すこ

とができる．なお，ここで TTP が受け取ったエスクローは，**上記のリストが空になった場合のみ** (つまり，すべての参加者間の問題が解消された場合のみ) 対応する受信者 P_j へ送付される．

フェーズ (4) において P_i から復号シェアを受け取れなかった参加者 (P_j とする) がいた場合，時刻 t_1 以降に，P_j はフェーズ (3) で入手している P_i から受け取ったエスクローを TTP に送信することで，TTP にその複号を依頼し対応する復号シェアを受け取ることができる．なお，TTP はこの処理を**上記のリストが空である場合のみ**実施しそれ以外の場合は依頼を無視する．直感的には，これは一部の交換のみが達成されることを防ぐための措置である．(MFE の安全性はすべての交換が達成されるか，すべてされないかのどちらか一方であることを保証するものであることに注意する．)

本研究の成果を理解するうえで重要な点として，KK-MFE ではフェーズ (2), (3) において，各参加者は自身が受け取るべきメッセージをすべて受け取ったら，**他参加者の完了や TTP の設定時間を待たず次のフェーズに進んでよい**としている．

3.2 例：3 者間の場合

本節では，KK-MFE の具体的な手順について説明する．簡単のため，図 1 の交換トポロジーに基づく 3 者間の場合に注目する．ここでは，すべての参加者が正直な場合のみを扱う．不正な参加者が存在する場合の手順については，3.3 節で説明する．以降，$\mathrm{lbl} := id||\Upsilon||t_1||t_2||\mathsf{pp}||v$ とする．なお，KK-MFE では複数のプロトコルで一つの TTP を共有して使用することを想定しており，lbl は TTP がプロトコルを一意に特定するための識別子として用いる．

(1) セットアップ

TTP は $\mathsf{PkGen}(1^l) \to (\mathsf{SK}_\mathsf{T}, \mathsf{PK}_\mathsf{T})$ を実行し PK_T を公開する．3 人の参加者らで $\mathsf{ThGen}(1^l, 3, 3) \to (\mathsf{pp}, v, \{x_i\}_{1\le i\le n})$ を実行し，参加者 P_i が x_i を受け取り，(pp, v) を公開値とする．

(2) 暗号化アイテム交換フェーズ

i. $i \in [3]$ について，P_i は $\mathsf{ThPrvEnc}(\mathsf{pp}, f_i, \gamma_i) \to (\mathsf{VE}_i, \mathsf{VEproof}_i)$ を実行する．その後，P_i は生成した $(\mathsf{VE}_i, \mathsf{VEproof}_i)$ を他 2 人へ送信する．

ii. 各 P_i は，すべての $j \in [3]$ について $\mathsf{ThVrfy}(\mathsf{pp}, \gamma_j, \mathsf{VE}_j, \mathsf{VEproof}_j)$ を実行し，検証結果がすべて T であることを確認する．そうでなければ，プロトコルを中止する．このときの通信の様子を図 2 に示す (なお，この図の中では証明の値については省略する．これは後述の図 3–4 でも同様である)．

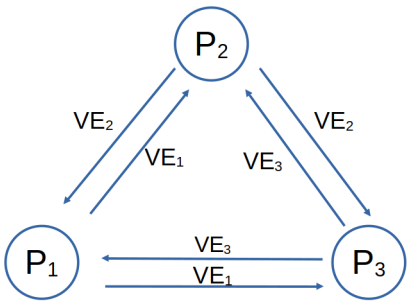

図 2 フェーズ (2) での通信

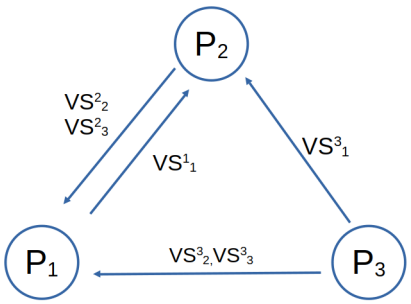

図 3 フェーズ (3) での通信

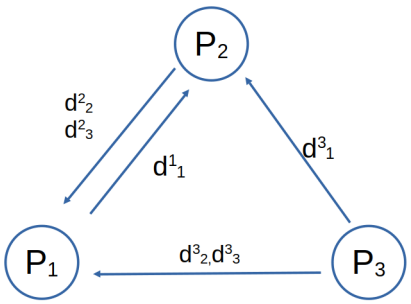

図 4 フェーズ (4) での通信

(3) 復号鍵シェアエスクロー交換フェーズ

i. 各 P_i はすべての $j \in [3]$ について $\mathsf{ThDShr}(x_i, \mathsf{pp}, \mathsf{VE}_j) \to (d_j^i, \mathsf{DSproof}_j^i)$ を実行する．

ii. 各 P_i はすべての $j \in [3]$ について $\mathsf{PkPrvEnc}(\mathsf{PK}_\mathsf{T}, (d_j^i, \mathsf{DSproof}_j^i), \psi_j; \mathrm{lbl}) \to (\mathsf{VS}_j^i, \mathsf{VSproof}_j^i)$ を実行する．なお，ψ_j は $(\mathsf{pp}, \mathsf{VE}_j, v)$ に基づいて $d_j^i||\mathsf{DSproof}_j^i$ が VE_j の復号鍵のシェアとその証明であることを検証する述語である．

iii. P_1 は P_2 に $(\mathsf{VS}_1^1, \mathsf{VSproof}_1^1)$ を送る．P_2 は P_1

に $(\mathsf{VS}_2^2, \mathsf{VSproof}_2^2, \mathsf{VS}_3^2, \mathsf{VSproof}_3^2)$ を送る．P_3 は P_1 に $(\mathsf{VS}_1^3, \mathsf{VSproof}_1^3, \mathsf{VS}_3^3, \mathsf{VSproof}_3^3)$ を，P_2 に $(\mathsf{VS}_2^3, \mathsf{VSproof}_2^3)$ を送る．このときの通信の様子を図 3 に示す．

iv. 各 P_i は受信したすべての $(\mathsf{VS}_k^j, \mathsf{VSproof}_k^j)$ を PkVrfy で検証する．例えば，P_2 は $\mathsf{PkVrfy}(\mathsf{VS}_1^1, \mathsf{VSproof}_1^1)$ と $\mathsf{PkVrfy}(\mathsf{VS}_1^3, \mathsf{VSproof}_1^3)$ を実行し，検証結果がすべて T であることを確認する．ここで，検証結果が F となる $(\mathsf{VS}_k^j, \mathsf{VSproof}_k^j)$ があった場合は，その j について TTP を呼び出してアルゴリズム 2 を実行する (詳細は 3.3 節で説明する)．

(4) 復号鍵シェア交換フェーズ

i. P_1 は P_2 に $(d_2^1, \mathsf{DSproof}_2^1)$ を送る．P_2 は P_1 に $(d_2^2, \mathsf{DSproof}_2^2, d_3^2, \mathsf{DSproof}_3^2)$ を送る．P_3 は P_1 に $(d_2^3, \mathsf{DSproof}_2^3, d_3^3, \mathsf{DSproof}_3^3)$ を，P_2 に $(d_1^3, \mathsf{DSproof}_1^3)$ を送る．このときの通信の様子を図 4 に示す

ii. 各 P_i は受信したすべての $(d_k^j, \mathsf{DSproof}_k^j)$ を PkVrfy で検証する．例えば，P_2 は $\mathsf{PkVrfy}(d_1^1, \mathsf{DSproof}_1^1)$ と $\mathsf{PkVrfy}(d_1^3, \mathsf{DSproof}_1^3)$ を実行し，検証結果がすべて T であることを確認する．ここで，検証結果が F となる $(d_k^j, \mathsf{DSproof}_k^j)$ があった場合は，その j について TTP を呼び出してアルゴリズム 4 を実行する (詳細は 3.3 節で説明する)．

iii. 要求するアイテムのある各参加者 P_i は $\mathsf{ThDec}(\{d_k^j\}_{1\le j\le n}, \mathsf{pp}, \mathsf{VE}_k) \rightarrow f_k$ で所望のアイテム f_k を復号して得る．なお，P_3 のように，要求するアイテムのない参加者はこの手順を行わない．

3.3 KK-MFE の安全性

以下に，TTP の役割を踏まえて KK-MFE 方式の安全性の直感的な理解を与える．

Resolve 1 (アルゴリズム 2). 前節の手順 (3)-iv で，検証結果が F となる $(\mathsf{VS}_k^j, \mathsf{VSproof}_k^j)$ があった場合に，対応する交換トポロジーにおけるエッジ (つまり，誰と誰の間で問題が生じたか) の情報を記録するためのアルゴリズムである．その記録は DS にされ，$\mathrm{DS}[u,j] = \perp$ であることは P_j が送るアイテム f_u について問題が報告されたことを意味する．なお，DS, complaintList が更新されるたび，その内容がすべての参加者らへ通知されるものと仮定する．

Resolve 2 (アルゴリズム 3). Resolve 1 で問題が生じているとされている $\mathrm{DS}[u,j] = \perp$ となる (u,j) について，(u,j) に対応する暗号文のエスクローを持つ P_i(Resolve 2 の実行者が問題を報告された本人である場合 $i = j$) が対応する (本来手順 (3)-iii で送るべきであった) 暗号文のエスクロー $(\mathsf{VS}_u^j, \mathsf{VSproof}_u^j)$ を TTP に送信することで問題を解消するためのアルゴリズムである．呼び出された TTP は，PkDec で復号した $(d_u^j, \mathsf{DSproof}_u^j)$ を ThDSVrfy で正当性を検証し，検証結果が T であった場合は $\mathrm{DS}[u,j]$ を $(d_u^j, \mathsf{DSproof}_u^j)$ で上書きする．これは (u,j) の問題が解消されたことを意味する．ここで得た $(d_u^j, \mathsf{DSproof}_u^j)$ は，Resolve 3 の中で，対応する受信者へ送信される．

Resolve 3 (アルゴリズム 4). $\mathrm{DS}[u,j] = \perp$ となる (u,j) が存在しない場合 (つまり，問題が発生していないか，すべて解消された場合) のみ，実行可能である．フェーズ (4) で復号鍵のシェアを受け取れなかった参加者が，そのシェアに対応する暗号文 (エスクロー) を TTP に送信することで，その複号を依頼するためのアルゴリズムである．また，Resolve 2 で解消された問題の復号鍵のシェア $\mathrm{DS}[u,j] = (d_u^j, \mathsf{DSproof}_u^j)$ を TTP が保持する場合は，その対応する受信者からの呼び出し ($\mathcal{V} = 0$ として呼び出す) で $(d_u^j, \mathsf{DSproof}_u^j)$ を返信する．

4. 提案攻撃手法

本節では KK-MFE において，2 人以上の参加者を corrupt した攻撃者が，正直な参加者のアイテムを持ち逃げできることを示す．つまり，KK-MFE が 2.4 節で与えた安全性を満たさないことを示す．

4.1 攻撃手法の概要

提案の攻撃手法では，KK-MFE における以下 2 つの性質を利用する．

A. フェーズ (3) で受信すべきメッセージをすべて受け取った参加者は，他参加者のそのフェーズの完了や TTP の設定時間 t_1, t_2 を待たず次のフェーズ (4) に進んでよい．

B. Resovle 3 は，$\mathrm{DS}[u,j] = \perp$ となる (u,j) が存在しない場合のみ実行可能である．

性質 A より，正直な参加者が t_1 以前にフェーズ (4) を開始する場合が生じ得ることに注意する．[*2] このとき，ある正直な参加者がフェーズ (4) を完了した後に，Resolve 1 が実行され DS のある項目に $\perp$ が書き込まれる事象が発生しうる (さらに，Resolve 2 による問題の解消も行われない)．したがって，その正直な参加者がフェーズ (4) で，他参加者から復号鍵のシェアを受け取れなかった場合に Resolve 3 を呼び出したとしても，性質 B より TTP はそれに応じずアイテムを得られない．しかし，その正直な参加者はすでにフェーズ (4) を実行済みであるため，悪意ある参加者はその正直な参加者からのアイテムを得ることができる．

[*2] Kılınç ら [9] は，時間のパラメータ t_1, t_2 はすべての参加者が正直であればプロトコルに影響を与えないことを踏まえて，これらの値は十分に余裕を持った値に設定にしてよいと言及している．実際，彼らは t_1 を数時間から数日に設定する例を挙げている．

これは MFE の安全性に反する状況である．具体的な攻撃手順を次節に示す．

4.2 手順

図 1 において，P_2 と P_3 が不正な参加者である場合を考える．P_2 と P_3 はフェーズ (3) までプロトコルに従い動作するが，フェーズ (4) は P_1 がフェーズ (4) を完了するまで行わずに待機する．ここで，P_1 がフェーズ (4) を時刻 t_1 以前に完了したと仮定する．フェーズ (4) の P_1 からのメッセージをすべて受信したのち，P_2 と P_3 は以下の手順を行う．

i. P_2 は VS_1^3 についてアルゴリズム 2 を実行し，$\mathrm{DS}[1,3] = \perp$ を TTP に記録させる．なお，P_3 はこの問題の解消を行わない．

ii. P_3 は P_2 へフェーズ (4) のメッセージとして $(d_1^3, \mathsf{DSproof}_1^3)$ を送信するが，P_1 へは送らない．

iii. P_1 は VS_1^3 についてアルゴリズム 4 を実行し復号を依頼するが，$\mathrm{DS}[1,3] = \perp$ であるためこの依頼は無視される．

iv. その結果，不正な P_2 はアイテム f_1 を得るが，正直な P_1 は f_2 を得られない．

注意 1. アルゴリズム 3 の Input に着目すると，P_j に関するアイテム f_u の復号鍵シェア VS_u^j について問題 $\mathrm{DS}[u,j] = \perp$ は，P_j 以外の参加者が解消することも可能である．したがって，P_1 がアルゴリズム 3 で $\mathrm{DS}[1,3] = \perp$ を解消することができれば，上述の手順 iii が成り立たず，提案攻撃手法は成功しないことになる．しかし，図 3 からわかるようにエスクローの送信はトポロジーに依存しており，$\mathrm{DS}[u,j] = \perp$ を解消するために必要な VS_1^3 を P_1 は得られない．したがって，P_1 がアルゴリズム 3 で $\mathrm{DS}[1,3] = \perp$ を解消することはできない．

注意 2. 提案手法で攻撃が成功する場合は，交換トポロジーにおいて正直な参加者から不正な参加者へのエッジがある場合に限られる．よって，図 1 の P_3 のようなアイテムを受け取らない参加者のみで結託したとしても，今回の手法は適用できない．

5. 提案方式: 提案攻撃への対策

本節で，4 節で示した問題を解消する (つまり，攻撃を防ぐ) 手法を 2 つ提案する．なお，両手法において，TTP のアルゴリズムは KK-MFE と同様であり変更は加えない．

5.1 提案手法 1

1 つ目の手法は，4 節で示した攻撃手法が「正直な P_1 が時刻 t_1 より前にフェーズ (4) を開始し得る」ことを前提としている点を利用する．これが成立しないようにするため，正直な参加者のフェーズ (4) を開始する条件として以下 2 つを加える．

- $t_{now} > t_1$
- $\forall (x,y) \in I \times [n], \mathrm{DS}[x,y] \neq \perp$

攻撃者は時刻 t_1 以降に DS を更新することはできない (つまり、アルゴリズム 2 を実行できない) ため，この変更により 4.2 節の手順 iii のような状況が発生することがなくなり，提案の攻撃手法は適用できない．なお，この修正手法では，すべての参加者が正直である場合を含めて，プロトコルの実行時間が t_1 に依存する．

5.2 提案手法 2

2 つ目の手法は，注意 1 での考察を利用する．つまり，P_1 が $\mathrm{DS}[1,3] = \perp$ を解消することができれば，上述の手順 iii が成り立たず提案の攻撃手法は適用できなくなる．これを実現するため，フェーズ (3) において，エスクローを対応するアイテムを要求する参加者へのみ送信していた手順を，すべての参加者へ送信する手順へ変更する．つまり，3.1 節の (3) を以下のように変更する．

(3) 復号鍵シェアエスクロー交換: すべての $i \in I$ について，各参加者 P_k は ThDShr で VE_i の復号鍵のシェア d_k^i を生成し，さらに $\mathsf{PkPrvEnc}_{\mathsf{PK_T}}$ で d_k^i を暗号化した暗号文 VS_k^i を生成する．その後，$\Upsilon[k,i] \neq 1$ であるすべての i について VS_k^i を**すべての参加者**へ送信する．

なお，3.2 節の手順 (3)-iii は以下のように変更される．

iii. P_1 は P_2, P_3 に $(\mathsf{VS}_1^1, \mathsf{VSproof}_1^1)$ を送る．P_2 は P_1, P_3 に $(\mathsf{VS}_2^2, \mathsf{VSproof}_2^2, \mathsf{VS}_3^2, \mathsf{VSproof}_3^2)$ を送る．P_3 は P_1, P_2 に $(\mathsf{VS}_1^3, \mathsf{VSproof}_1^3, \mathsf{VS}_2^3, \mathsf{VSproof}_2^3, \mathsf{VS}_3^3, \mathsf{VSproof}_3^3)$ を送る．このときの通信の様子を図 5 に示す．

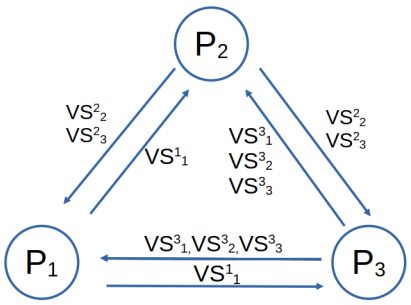

図 5 提案手法 2 のフェーズ (3) での通信

アルゴリズム 3 で，P_j に関する問題 $\mathrm{DS}[u,j] = \perp$ は，

P_j 以外の参加者が解消することも可能であることに注意する．提案手法 2 による手順の変更で，P_1 は VS_1^3 を受け取らない限りフェーズ (4) に進まないこととなる．したがって，4.2 節の手順 i のようなふるまいを攻撃者が行ったとしても，フェーズ 4 では P_1 は VS_1^3 を保持していることが保証される．P_1 はアルゴリズム 3 を VS_1^3 について実行することで $\mathrm{DS}[1,3] = \perp$ を解消することが可能であり，手順 iii のような状況を回避できる．

上述の $\Upsilon[k,i] \neq 1$ の条件より，各参加者 P_j は $\Upsilon[i,j] = 0$ に関する f_i (つまり，P_j が要求していないアイテム) に対応する復号鍵のシェアを高々 $n-1$ 個までしか得られない．シェアの閾値は n であることに注意する．したがって，提案手法 2 での手順の変更を加えてもアイテムの秘匿性は失われない (任意の参加者は，自身が交換トポロジーで要求するアイテム以外の情報は得られない)．実際，図 5 において，例えば P_3 はフェーズ (3) 開始時点で f_1 に関するシェアとして VS_1^3 を持ち，f_2 に関するシェアとして VS_2^3 を持つ．その後，フェーズ (3) の通信で P_3 は f_1 のシェア VS_1^1 と，f_2 のシェア VS_2^2 を受け取るが，$\mathsf{VS}_1^2, \mathsf{VS}_2^1$ はどちらも受け取らない．したがって，TPKE の秘匿性より，P_3 は f_1, f_2 に関する情報を何も得られない．P_2 についても同様の議論が成立する．(交換トポロジーより，P_1 はすべてのアイテムを正規に得られることに注意する．)

6. おわりに

本稿では，Kılınç と Küpçü [9] によって提案されたマルチパーティ公平交換プロトコルに対する攻撃手法とその対策の提案を行った．提案の攻撃手法では，参加者を 2 人以上 corrupt した攻撃者が，正直な参加者のアイテムを持ち逃げできることを示した．さらに，提案攻撃を防ぐ手法を 2 つ提案したが，形式的な安全性証明は与えていない．したがって，今後の研究課題として，5 節で示した提案手法について形式的な安全性証明を与えることがある．

謝辞 本研究は JSPS 科研費 JP23K16880, JST CREST JPMJCR22M1 および JST CRONOS Japan Grant Number JPMJCS24K2 の助成を受けたものです．

参考文献

[1] N. Asokan, Matthias Schunter, and Michael Waidner. Optimistic protocols for fair exchange. In *Proceedings of the 4th ACM Conference on Computer and Communications Security*, CCS '97, page 7–17. Association for Computing Machinery, 1997.

[2] Alptekin Küpçü and Anna Lysyanskaya. Usable optimistic fair exchange. *Comput. Netw.*, 56(1):50–63, 2012.

[3] Henning Pagnia and Felix C. Gartner Darmstadt. On the impossibility of fair exchange without a trusted third party. 1999.

[4] Christian Cachin and Jan Camenisch. Optimistic fair secure computation. In *Annual International Cryptology Conference*, pages 93–111. Springer, 2000.

[5] N. Asokan, Matthias Schunter, and Michael Waidner. *Optimistic protocols for multi-party fair exchange*. IBM TJ Watson Research Center, 1996.

[6] Feng Bao, Robert Deng, Khanh Quoc Nguyen, and Vijay Varadharajan. Multi-party fair exchange with an off-line trusted neutral party. In *Proceedings. Tenth International Workshop on Database and Expert Systems Applications. DEXA 99*, pages 858–862. IEEE, 1999.

[7] Nicolás González-Deleito and Olivier Markowitch. An optimistic multi-party fair exchange protocol with reduced trust requirements. In Kwangjo Kim, editor, *Information Security and Cryptology — ICISC 2001*, pages 258–267. Springer Berlin Heidelberg, 2002.

[8] Insoo Khill, Jiseon Kim, Ingoo Han, and Jaecheol Ryou. Multi-party fair exchange protocol using ring architecture model. *Computers & Security*, 20(5):422–439, 2001.

[9] Handan Kılınç Alper and Alptekin Küpçü. Optimally efficient multi-party fair exchange and fair secure multi-party computation. *ACM Transactions on Privacy and Security*, 25(1):1–34, 2021.

[10] Victor Shoup and Rosario Gennaro. Securing threshold cryptosystems against chosen ciphertext attack. *Journal of Cryptology*, 15(2):75–96, 2002.

[11] Jan Camenisch and Ivan B Damgård. Verifiable encryption and applications to group signatures and signature sharing. *BRICS Report Series*, 5(32), 1998.