公開鍵暗号を用いた加法的乱択符号化の効率的な構成

吉村 昌也[1,a)]　淺野 京一[1,2]　岩本 貢[1]　渡邉 洋平[1,2]

概要：
Halevi ら（CRYPTO 2023）によって提案された加法的乱択符号化 (Additive Randomized Encoding: ARE) は，n 人の参加者がそれぞれ n 入力の関数 f に対する入力 x_i を（乱択）符号化し，それらの和を復号することで，各参加者の入力のプライバシーを保証しつつ $f(x_1, \ldots, x_n)$ を評価できる暗号技術である．Bitansky ら (CRYPTO 2025) は，ARE を弱めた概念として，特定の参加者の入力のプライバシーのみを保証する One-Sided ARE (OSARE) を新たに提案し，公開鍵暗号のみを仮定したうえで，2-party OSARE から n-party ARE への一連の変換手法を示した．本論文では，Bitansky らの変換手法内で用いる OSARE を再考することで，最終的により効率的な n-party ARE 方式が得られることを示す．

キーワード：加法的乱択符号化，マルチパーティ計算，公開鍵暗号

Efficiency Improvement of Additive Randomized Encodings from PKE

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

Abstract: Halevi et al. (CRYPTO 2023) introduced additive randomized encoding (ARE), a cryptographic primitive in which each of the n-parties encodes their input x_i to an n-party function f using randomized encoding; by decoding the sum of the encodings, one can evaluate $f(x_1, \ldots, x_n)$ while preserving the privacy of each party's input. Bitansky et al. (CRYPTO 2025) introduced one-sided ARE (OSARE), a weaker notion that preserves only the privacy of a designated party's input. They proposed a transformation which lifts 2-party OSARE to n-party ARE by only assuming public-key encryption. In this paper, we revisit the OSARE construction used in Bitansky et al.'s transformation, and then we can obtain a more efficient n-party ARE scheme.

Keywords: Additive Randomized Encoding, Multi-Party Computation, Public-Key Encryption

1. はじめに

1.1 研究背景

マルチパーティ計算 (Multi-Party Computation: MPC) とは，n 人の参加者が協力し，自身の入力 x_i を他者に秘匿しながら計算及び通信を行い，最終的に $f(x_1, \ldots, x_n)$ を計算する暗号技術である [11]．非対話 MPC (Non-Interactive MPC: NIMPC) は，参加者による自身の入力に関するメッセージの送信が 1 回に制限された MPC であり，参加者とは別に評価者を設定し，各参加者は自身の入力と事前に共有されたパラメータに応じたメッセージを 1 回限り評価者に送信することで，評価者は関数を評価できる [4]．NIMPC では参加者同士の対話がないため，参加者がプロトコルに同時に参加できない状況に適しており，電子投票やオークションなどにおいて有用である．

シャッフルモデルにおける NIMPC. Ishai ら [8] は，異なる当事者からのメッセージがシャッフルされるシャッフルモデルにおいて，加算演算に対する NIMPC を提案した．具体的には，各参加者は加法的秘密分散法 [2], [10] を用いて自身の入力をシェアに分散し，そのシェアを評価者に送

[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
National Institute of Advanced Industrial Science and Technology
[a)] yoshimura@uec.ac.jp

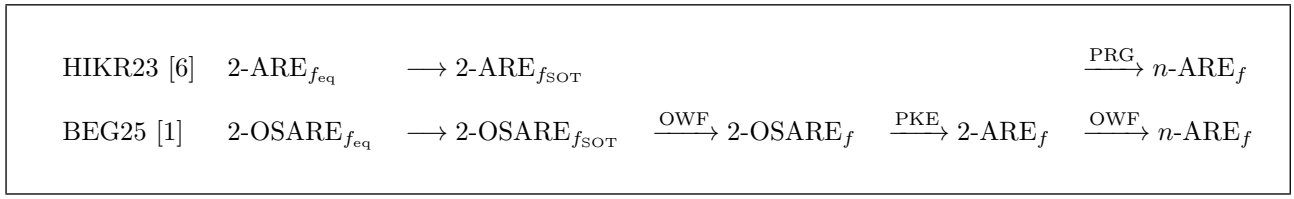

図 1: Base-ARE から任意の関数 f に対する ARE の構成法
Fig. 1 Constraction of ARE for an arbitrary function f from Base-ARE

信する．評価者は，シェアのメッセージをすべて加算することで，各参加者の入力の合計値を復元することができる．ここで，評価者に送られるシェアはシャッフルされているため，評価者が，受け取ったシェアの送信元を特定して送信元の入力を復元するようなことは困難である．また，このようなシャッフルモデルは匿名通信によって実現できることが知られている

Halevi ら [6] は，シャッフルモデルにおいて任意の関数 f における NIMPC を実現した．彼らは，任意のマルチパーティ関数の計算を，単純な演算であるアーベル群上の加算のみで計算可能な加法的乱択符号化 (Additive Randomized Encoding: ARE) を提案した．具体的には，n 入力関数 f に対する ARE において，n 人の参加者がそれぞれ n 入力の関数 f に対する入力 x_i を（乱択）符号化し，それらの和を復号することで，各参加者の入力のプライバシーを保証しつつ $f(x_1, \ldots, x_n)$ を評価できる．Halevi らは，任意の関数 f に対する ARE に Ishai らのプロトコルを組み合わせることで，シャッフルモデルにおける任意の関数 f の NIMPC の構成を初めて得た．

ARE の構成法． Halevi らは，Additive-Squaring Diffie–Hellman (ASDH) 仮定[*1]に基づき，等号判定関数 f_{eq} に対する ARE 方式を提案した．また，任意の 2 者間の等号判定関数 f_{eq} に対する ARE 方式から文字列紛失通信 (String-Oblivious Transfer: SOT) [5], [7], [9] に対応した関数 f_{SOT} に対する 2 者間の ARE 方式への変換，及び，それを任意の関数 f に対する ARE 方式に変換できることを示し，任意の関数 f に対する 2 者間の ARE 方式を並列に並べることで，f に対する n 者間の ARE 方式に変換できることを示した (図 1)．すなわち，ASDH 仮定及び疑似乱数生成器 (Pseudorandom Generator: PRG) に基づく ARE 方式が提案された[*2]．ここで，任意の関数に対する ARE の構成にはガーブルドサーキット (Garbled Circuit：GC) を用いているため，f_{SOT} に対する ARE の構成を必要としている．

その後，Bitansky ら [1] は，Computational Diffie–Hellman (CDH) 仮定に基づく 2 者間の f_{eq} に対する ARE 方式を提案し，結果としてペアリングを用いない初めての任意の関数 f に対する ARE 方式を得た．また，Bitansky らは，公開鍵暗号 (Public-Key Encryption: PKE) [3] から ARE が構成できることを示した．具体的には，2 者間 ARE において片側の入力のみが安全である One-Sided ARE (OSARE) を新たに提案し，任意の関数 f に対する統計的識別不可能な OSARE 方式を PKE を用いて ARE 方式に変換する手法を示した (図 1 参照)．また，f に対する統計的識別不可能な OSARE は，2 者間の統計的識別不可能な f_{eq} に対する OSARE 方式を具体的に構成し，Halevi らの ARE 方式の変換が OSARE に対しても用いることが可能であることを示すことで得られる．

統計的識別不可能な f に対する OSARE によって，PKE の計算量仮定に依存した ARE が構成できることが示されたが，その OSARE を得るには変換の過程で通信複雑性が増加してしまう．具体的には，Bitansky らによって示された f_{eq} に対する OSARE の通信複雑性はセキュリティパラメータ $O(\lambda)$ に比例し，図 1 にあるように，いくつかの変換を経た後，最終的に得られる f に対する ARE の通信複雑性は $O(\lambda^2)$ に比例してしまう．これは，具体的な ARE の構成のセキュリティパラメータ λ に関する通信複雑性が $O(\lambda)$ で得られることに対して，非効率的である．

1.2 本研究の貢献

本研究では，PKE を用いた任意の関数に対する ARE のより効率的な構成法を示す．提案する構成法によって最終的に得られる，任意の関数 f に対する ARE の通信複雑性は，$O(\lambda n \ell)$ である．ここで，λ はセキュリティパラメータ，n は参加者数，ℓ は関数 f の入力長である．

Halevi らと Bitansky らは最終的に GC によって任意の関数に対する ARE を構成しており，そのために紛失通信 (String Oblivious Transfer: SOT) の関数 f_{SOT} に対する ARE（または OSARE）を用いている．任意の関数 f に対する ARE において，f を計算する回路は GC によって暗号化され，暗号化に用いた各ユーザに入力の 1 ビットごとに対応したラベルを，f_{SOT} に対する ARE によって評価者に送信する．ラベル長は λ に依存するため，参加者 n 人の入力 ℓ ビットのラベル長を受け取るため評価者が受け取る情報量は $O(\lambda n \ell)$ である．そのため，（GC を用いて実現す

[*1] ASDH 仮定は Square External Diffie–Hellman (Square XDH) 仮定に帰着することが知られている [6].
[*2] Håstad ら [7] によって一方向性関数 (One-Way Function: OWF) と PRG の等価性が示されていることから，OWF に基づくと言い換えることもできる．ここでは Halevi ら [6] の記述に従い PRG としている．

表 1: ARE の構成法の効率性比較.

Table 1 Efficiency comparison of ARE constructions.

方式	2-party Base-(OS)ARE			変換に必要な計算量仮定	通信複雑性
	関数	計算量仮定	ペアリング		
HIKR23 [6]	f_{eq}	ASDH	✓	PRG	$2\lambda\ell(n-1)(\log\lvert\mathbb{G}_1\rvert+\log\lvert\mathbb{G}_2\rvert+\log\lvert\mathbb{G}_T\rvert)+O(\lambda\lvert C_{\lambda,n,\ell}\rvert)$
BEG25 [1]	f_{eq}	CDH	–	OWF	$2\lambda\ell(n-1)(4\log\lvert\mathbb{G}\rvert+1)\cdot O(\lambda)+O(\lambda\lvert C_{\lambda,n,\ell}\rvert)$
BEG25 [1]	f_{eq}	–	–	PKE, OWF	$4\lambda^2(\ell_g+\ell_e)(n-1)+O(\lambda\lvert C_{\mathrm{sFE}_g}\rvert)+O(\lambda\lvert C_{\mathrm{sFE}_e}\rvert)$
提案方式（§4.1）	f_{SOT}	–	–	PKE, OWF	$(2\lambda+1)(\ell_g+\ell_e)(n-1)+O(\lambda\lvert C_{\mathrm{sFE}_g}\rvert)+O(\lambda\lvert C_{\mathrm{sFE}_e}\rvert)$

る限り）ARE の通信複雑性は $O(\lambda n\ell)$ であることが理想的であると考えられるものの，Bitansky らによる PKE を用いた ARE の構成は通信複雑性が $O(\lambda^2 n\ell)$ である．

Bitansky らは，f_{SOT} に対する OSARE を，f_{eq} に対する OSARE を用いて構成した．しかし，彼らが提案した f_{eq} に対する OSARE 方式の通信複雑性は $O(\lambda)$ である．また，入力は任意であるが，出力は 1 ビットに制限された任意の関数 f_{small} に対する OSARE の構成は，f_{eq} に対する OSARE 方式から構成でき，出力に λ ビットを要求する f_{SOT} に対する OSARE を構成するためには，$O(\lambda^2)$ の通信複雑性を必要としていた．

そこで本研究では，f_{SOT} に対する OSARE を直接構成することで，Bitansky らの変換法で生じるオーバーヘッドを回避し，結果として PKE を用いた効率的な ARE を得る．提案する f_{SOT} に対する OSARE の通信複雑性は $O(\lambda)$ であるため，最終的に得られる PKE を用いた任意の関数に対する ARE の通信複雑性は $O(\lambda n\ell)$ となる．

効率性の比較. 表 1 は提案方式と既存方式を比較したものである．表中の Base-(OS)ARE とは，各手法による変換において最も基本的な要素技術となる (OS)ARE のことを表す．f_{eq}，f_{SOT} はそれぞれ等号判定関数と SOT 関数を表すものとする．g, e を変換に用いられる単一鍵関数型暗号に関連した関数とし，ℓ_g, ℓ_e をそれぞれの入力長，$C_{\mathrm{sFE}_g}, C_{\mathrm{sFE}_e}$ をそれぞれの回路表現とする．詳細は [1] を参照されたい．

Halevi らや Bitansky らが Base-(OS)ARE として，等号判定関数 f_{eq} に対する (OS)ARE を用いていることに対して，提案方式の Base-OSARE は f_{SOT} に対する OSARE である点が既存方式と異なる．また，表の上 2 つの方式は，ある計算量仮定（1 つめはペアリングが必要な ASDH 仮定であり，2 つめはペアリングが不要な CDH 仮定である）から Base-ARE を構成しており，PRG または OWF の仮定から最終的に ARE が構成される．一方で，表の下 2 つの方式は，統計的識別不可能な（すなわち計算量仮定を必要としない）Base-OSARE から構成されている．Base-OSARE は情報理論的に実現することができるため，任意の PKE（PKE の計算量仮定）から ARE が構成される．また，セキュリティパラメータを λ とし，最終的に得られる，参加者数 n，入力長 ℓ の f に対する ARE の通信複雑性に関しては，表の 4 つの方式に共通して，$O(\ell n)$ に比例すること．および，GC による f（OSARE の場合は変換に必要な関数 g, e）の回路サイズに依存することが挙げられる．一方異なる点として，上記 2 つの Based-ARE の λ に関する通信複雑性は $O(\lambda)$ であるに対し，Bitansky らが提案した Base-OSARE による ARE の構成の通信複雑性は $O(\lambda^2)$ である．それに対し，Base-OSARE による提案方式の ARE の構成の通信複雑性は $O(\lambda)$ である．

2. 準備

2.1 記法

λ はセキュリティパラメータを表す．有限集合 S に対して，S から一様ランダムに要素 s を取り出すことを $s \xleftarrow{\$} S$ と表記する．任意の自然数 $m, n \in \mathbb{N}$ に対し，$[m, n] \coloneqq \{m, m+1, \ldots, n\}$ とする．ただし $m = 1$ のとき，単に $[n]$ と表記する．

2.2 加法的乱択符号化

定義 2.1 (加法的乱択符号化 [6])**.** 関数 $f\colon (\{0,1\}^*)^* \to \{0,1\}^*$ をマルチパーティ関数とする．f に対する ARE 方式 S_f は 3 つのアルゴリズム $S_f = (\mathsf{Setup}, \mathsf{Enc}, \mathsf{Dec})$ からなる．

- $\mathsf{Setup}(1^\lambda, 1^n, 1^\ell) \to \mathsf{pp}$: 多項式時間のアルゴリズムで，セキュリティパラメータ λ，参加者の数 n， マルチパーティ関数 f の入力長 ℓ を入力に取り，公開パラメータ pp を出力する．公開パラメータには，λ, n, ℓ およびアーベル群 $\mathbb{G}$ の明示的な記述を含む．
- $\mathsf{Enc}(\mathsf{pp}, i, x_i) \to \hat{x}_i$: 多項式時間のアルゴリズムで，パーティ i の入力 X_i を入力に取り，$\hat{x}_i \in \mathbb{G}$ を出力とする．
- $\mathsf{Dec}(\mathsf{pp}, \hat{z}) \to y$: 多項式時間のアルゴリズムで，$\hat{z} \in \mathbb{G}$

を入力に取り，z を出力する．

ϵ-正当性． $\epsilon = \epsilon(\lambda)$ として，任意の $\lambda, n, \ell \in \mathbb{N}$ および $x_n \in \{0,1\}^\ell$ に対して，$\mathsf{pp} \leftarrow \mathsf{Setup}(1^\lambda, 1^n, 1^\ell), \hat{x}_i \leftarrow \mathsf{Enc}(\mathsf{pp}, i, x_i), \hat{z} = \sum_{i=1}^n \hat{x}_i$ のとき，

$$\Pr[\mathsf{Dec}(\mathsf{pp}, \hat{z}) = f(x_1, \ldots, x_n)] \geq 1 - \epsilon(\lambda),$$

を満たし，ϵ は λ に対して無視可能であるとき，f に対する ARE 方式 S_f は ϵ-正当性を満たすという．

安全性． 多項式時間シミュレータ Sim が存在し，任意の $\lambda, n, \ell \in \mathbb{N}, x_1, ..., x_n \in \{0,1\}^\ell$ に対して，

$$\mathsf{Sim}(1^\lambda, 1^n, 1^\ell, f(x_1, ..., x_n)) \approx_c (\mathsf{pp}, \hat{z}),$$

を満たすとき，f に対する ARE 方式 S_f は計算量的に安全であるという．ただし，$\mathsf{pp} \leftarrow \mathsf{Setup}(1^\lambda, 1^n, 1^\ell), \hat{x}_i \leftarrow \mathsf{Enc}(\mathsf{pp}, i, x_i), \hat{z} = \sum_{i=1}^n \hat{x}_i$ である．

さらに，計算量的識別不可能性の代わりに統計的 / 完全識別不可能性がある場合，安全性は，統計的 / 完全に安全であるという．

n, ℓ が固定されている関数 f を考えるとき，上記の記法におけるこれらのパラメータを省略する．また，Setup アルゴリズムが不要の場合はそれを省略し，pp を直接明記する．

2.3 片側加法的乱択符号化

Bitansky, Erabelli, Garg らが上記の ARE の安全性より弱い概念である片側加法的乱択符号化 (One-Sided ARE: OSARE) を定義した [1]．OSARE は 2 パーティの ARE において，各パーティの各パーティの入力が符号化された値の和から，片側の入力のみ漏れないことが保証される．

定義 2.2 (One-Sided Additive Randomized Encodings [1])**.** 関数 $f : \{0,1\}^{\ell_x} \times \{0,1\}^{\ell_y} \to \{0,1\}^m$ を 2 者間の関数とする．f に対する ARE 方式 $S'_f = (\mathsf{Setup}, \mathsf{Enc}, \mathsf{Dec})$ が OSARE であるとは，通常の ARE の正当性を満たし，以下の安全性を満たすときである．

片側安全性． 多項式時間シミュレータ Sim が存在し，任意の $\lambda, \ell_x, \ell_y \in \mathbb{N}, x \in \{0,1\}^{\ell_x}, y \in \{0,1\}^{\ell_y}$ に対して，

$$\mathsf{Sim}(1^\lambda, x, f(x, y)) \approx_c (\mathsf{pp}, \hat{z}),$$

を満たすとき，f に対する OSARE 方式 S'_f は計算量的に安全であるという．ただし，$\mathsf{pp} \leftarrow \mathsf{Setup}(1^\lambda), \hat{x} \leftarrow \mathsf{Enc}(\mathsf{pp}, 1, x), \hat{y} \leftarrow \mathsf{Enc}(\mathsf{pp}, 2, y), \hat{z} = \hat{x} + \hat{y}$ である．

さらに，計算量的識別不可能性の代わりに統計的 / 完全識別不可能性がある場合，安全性は，統計的 / 完全に安全であるという．

2.4 1-out-of-2 紛失通信

定義 2.3 (紛失通信)**.** プロトコルは送信者 S と受信者 R の間の安全な 2 者間プロトコルとして定義される．ℓ ビットの文字列に対する 1-out-of-2 SOT は，以下のように定義される．

入力： S は文字列 $s_0, s_1 \in \{0,1\}^\ell$，$R$ は $c \in \{0,1\}$ を入力する．

出力： R は s_c $(c \in \{0,1\})$ を受け取る．

このとき SOT は，送信者 S が c の情報を一切得られず，また受信者 R が s_{1-c} の情報を一切得られないことを保証する．

3. 技術的概要

本節では，Bitansky らによる PKE を用いた ARE の構成法の枠組み [1] を説明し，本稿でその枠組みを改良するにあたって着目する点を示す．

3.1 Bitansky らによる公開鍵暗号を用いた ARE の構成

Bitansky らは，まず，OSARE を構成し，統計的識別不可能な 2 者間の等号判定関数 f_{eq} に対する OSARE を構成した．そして，Halevi ら [6] による変換を用いることで，出力が 1 ビットの任意の関数 f_{small} に対する 2 者間の OSARE へと変換できることを示した．この変換には，$f_{\mathrm{small}} : D_1 \times D_2 \to \{0,1\}, |D_1| \leq |D_2|$ としたとき，f_{eq} に対する OSARE に対して $|D_1|$ 倍の通信複雑性が増加する．

任意の関数 f に対する OSARE の構成には GC と SOT の関数 f_{SOT} に対する OSARE を用いる．λ ビットの SOT の関数 $f_{\mathrm{SOT}} : \{0,1\} \times \{0,1\}^{2\lambda} \to \{0,1\}^\lambda$ に対する OSARE は，f_{small} に対する OSARE を λ 回並列に実行すればよいため，f_{small} に対する OSARE の λ 倍の通信複雑性で f_{SOT} に対する OSARE 方式を得られる．このことから，$f_{\mathrm{SOT}} : \{0,1\} \times \{0,1\}^{2\lambda} \to \{0,1\}^\lambda$ であるので $(|\{0,1\}^{2\lambda}| > |\{0,1\}| = 2)$，$f_{\mathrm{eq}}$ に対する OSARE の通信複雑性に対して，2 倍の通信複雑性の f_{small} に対する OSARE が得られ，2λ 倍の通信複雑性の f_{SOT} に対する OSARE が得られる．

そして，Bitansky らは，2 者間の任意の関数 f に対する OSARE と PKE から 2 者間の f に対する ARE を得ることを示した．Halevi らが，パーティ $P_1, \ldots P_n$ に対して，P_1 と P_i $(i \in [2, n])$ の 2 者間 ARE を並列に行うことで n 者間 ARE を実現可能なことを示していることから，同様に n 者間の f に対する ARE を得られることを示した．

3.2 効率的な等号判定関数に対する OSARE の構成法の考察

3.1 節から分かるように，f_{eq} に対する OSARE の通信複

雑性に対して，通信複雑性が 2λ 倍の f_{SOT} に対する OSARE が得られるため，通信複雑性が λ に依存しない f_{eq} に対する OSARE を構成可能であれば，通信複雑性が $O(\lambda)$ の f_{SOT} に対する OSARE が構成可能であることが分かる．

しかしながら，本節では，通信複雑性が λ に依存しない f_{eq} に対する OSARE を構成が困難である直感を示す．そこで，例えば以下のようなシンプルな有限領域 $D = \{1 \dots d\}$ における f_{eq} に対する OSARE の構成を考える．

- $\mathsf{Enc}(1^\lambda, 1, x) \to \hat{x}$: $\hat{x} \coloneqq x$.
- $\mathsf{Enc}(1^\lambda, 2, y) \to \hat{y}$: $\hat{y} \coloneqq -y$..
- $\mathsf{Dec}(1^\lambda, \hat{z}) \to$ equal / not-equal: $\hat{z} = 0$ であれば equal，そうでなければ not-equal を出力する．

一見すると，ϵ-正当性を満たす通信複雑性が $O(\log |D|)$ の f_{eq} に対する OSARE が構成可能のように見える．しかし，この構成は OSARE の安全性を満たさない．$f_{\mathrm{eq}}(x, y) = $ not-equal のとき，直感的には $\hat{z} = x - y$ であるため，$\hat{z}$ から y に関する情報が漏洩しており，実際に $\mathsf{Sim}(1^\lambda, x, \text{not-equal})$ は $\hat{z} = x - y$ をシミュレートできない．

そこで，出力が not-equal のときに，$\hat{z}$ が一様ランダムになるような OSARE の構成を考えたい．そこで乱数 r を参加者に共有した以下の構成を考える．

- $\mathsf{Setup}(1^\lambda) \to \mathsf{pp}$: $r \xleftarrow{\$} [d], \mathsf{pp} \coloneqq r$.
- $\mathsf{Enc}(1^\lambda, 1, x) \to \hat{x}$: $\hat{x} \coloneqq H(x + r)$
- $\mathsf{Enc}(1^\lambda, 2, y) \to \hat{y}$: $\hat{y} \coloneqq -H(y + r)$
- $\mathsf{Dec}(1^\lambda, \hat{z}) \to$ equal / not-equal: $\hat{z} = 0$ であれば equal，そうでなければ not-equal を出力する．

ここで H をハッシュ関数 $H : [2d] \to \mathbb{Z}_q$ とする．一見，H の衝突確率 ϵ に対して，ϵ-正当性を満たす通信複雑性 $\log q$ の f_{eq} のための OSARE が構成できるように見える．実際に H がランダムオラクルであれば，$f_{\mathrm{eq}}(x, y) = $ not-equal のとき，$\hat{z}$ は乱数を振る舞うように見える．しかし，x, y を固定したとき，正当性を満たすために $\hat{z}$ は，r に関して確定的に決定される必要があるため，1 回の OSARE を行った後に Setup アルゴリズムを再び行う必要がある（または，n 回の f_{eq} に対する OSARE に対して，pp を n 倍用意する必要がある）．これは，f_{SOT} に対する OSARE が，f_{eq} に対する OSARE を λ 回要求していることからも効率的でない．

4. 等号判定関数に対する OSARE の効率的な構成

3.2 節で議論したように，通信複雑性が λ に依存しない統計的に識別不可能な等号判定関数 f_{eq} に対する OSARE の構成は難しい．

そこで本研究では，ARE と比較して OSARE に要求される安全性が弱いことに着目し，SOT の関数 f_{SOT} に対する OSARE の具体的構成が可能かを考える．具体的には，OSARE では 2 者間における片側の入力は漏洩してもよい，つまり，f_{SOT} において，文字列の受信側の選択ビット $b \in \{0, 1\}$ の入力，または送信側の 2 つの文字列 $(s_0, s_1) \in \{0, 1\}^{2\lambda}$ のどちらかが，評価者が得る値である符号の和から漏洩してもよい．そこでまず，提案構成法への準備として，以下の安全ではない f_{SOT} に対する OSARE の構成を考える．

- $\mathsf{Enc}(1^\lambda, 1, c) \to \hat{x}$: 選択ビット $c \in \{0, 1\}$ を入力とし，$\hat{x} \coloneqq (c, 0, 0)$ を出力する．
- $\mathsf{Enc}(1^\lambda, 2, (s_0, s_1)) \to \hat{y}$: 文字列 $s_0, s_1 \in \{0, 1\}^\lambda$ に対して，$\hat{y} \coloneqq (0, s_0, s_1)$ を出力する．
- $\mathsf{Dec}(1^\lambda, \hat{z}) \to z$: $\hat{z} = (\hat{z_0}, \hat{z_1}, \hat{z_2})$ を入力に取り，$\hat{z_0} = 0$ であれば $z = \hat{z_1}$，$\hat{z_0} = 1$ であれば $z = \hat{z_2}$ として，z を出力する．

上記のような，相手の入力に 0 をマスクする形となるような符号化，つまり，評価者に各参加者の入力をそのまま送るような OSARE を観察する．評価者は符号の和として (c, s_0, s_1) をそのまま受け取るため，選択ビット c に対して文字列 s_c を得ることができ，正当性を満たすことは明らかである．ここで，安全性を考えると，2.3 節の OSARE の安全性の定義から，片側のパーティ 1 の入力である c 及び，関数の出力である s_c から，符号の和 (c, s_0, s_1) をシミュレートできればよい．しかし，s_0, s_1 はパーティ 2 の入力であるため，c, s_c から s_{1-c} をシミュレートできず，上記の構成は安全性を満たさない．

4.1 SOT 関数に対する効率的な OSARE の構成法

上記の 4 節の構成では，選択ビット c によって選択されていない文字列 s_{1-c} がシミュレートできない点が，構成を実現する上での問題点であった．そこで，選択されていない文字列 s_{1-c} が乱数によってマスクされるような，2 者間 SOT の関数 f_{SOT} に対する OSARE の具体的な構成を提案する．関数 f_{SOT} は正確には以下のように定義される．

$$f_{\mathrm{SOT}} \colon \{0,1\} \times \{0,1\}^{2\lambda} \to \{0,1\}^\lambda,$$
$$f_{\mathrm{SOT}}(c, (s_0, s_1)) = s_c.$$

我々が提案する，上記の f_{SOT} に対する OSARE 方式 S_{SOT} は以下の通りである．

- $\mathsf{Enc}(1^\lambda, 1, c) \to \hat{x}$: パーティ 1 は選択ビット $c \in \{0, 1\}$ を入力とし，$u \xleftarrow{\$} \{0, 1\}^\lambda$ とする．$c = 0$ であれば，

$\hat{x} := (c, 0, u)$，$c = 1$ であれば，$\hat{x} := (c, u, 0)$ を出力する．

- $\mathsf{Enc}(1^\lambda, 2, (s_0, s_1)) \to \hat{y}$: パーティ 2 は文字列 $s_0, s_1 \in \{0,1\}^\lambda$ に対して，$\hat{y} := (0, s_0, s_1)$ を出力する．
- $\mathsf{Dec}(1^\lambda, \hat{z}) \to z$: $\hat{z} = (\hat{z}_0, \hat{z_1}, \hat{z_2})$ を入力に取り，$\hat{z_0} = 0$ であれば $z = \hat{z_1}$，$\hat{z_0} = 1$ であれば $z = \hat{z_2}$ として，z を出力する．

定理 1. 4.1 節で示す f_{SOT} に対する OSARE 方式 S_{SOT} は，定義 2.3 の正当性を満たす．

証明. 任意の $\lambda \in \mathbb{N}$ および $c \in \{0,1\}, s_0, s_1 \in \{0,1\}^\lambda$ に対して，$\hat{x} \leftarrow \mathsf{Enc}(1^\lambda, 1, c), \hat{y} \leftarrow \mathsf{Enc}(1^\lambda, 2, (s_0, s_1)), \hat{z} = \hat{x} + \hat{y}$ とする．$c = 0$ のとき $\hat{z} = (0, s_0, s_1 + u)$ で，$\mathsf{Dec}(1^\lambda, \hat{z}) \to s_0$ であり，$c = 1$ のとき $\hat{z} = (1, s_0 + u, s_1)$ で，$\mathsf{Dec}(1^\lambda, \hat{z}) \to s_1$ である．よって以下の式を満たすため，f_{SOT} に対する OSARE 方式 S_{SOT} は定義 2.3 の正当性を満たす．

$$\Pr[\mathsf{Dec}(1^\lambda, \hat{z}) = f(x, y)] = 1. \qquad \square$$

定理 2. 4.1 節で示す f_{SOT} に対する OSARE 方式 S_{SOT} は，定義 2.3 の統計的片側安全性を満たす．

証明. $\mathsf{Sim}(1^\lambda, c, f_{\mathrm{SOT}}(c, (s_0, s_1)) \approx (1^\lambda, \hat{z})$ を示す．すなわち，片側の入力である $\mathsf{Enc}(1^\lambda, 1, c)$ の入力 c と，f_{SOT} の出力から $\hat{z}$ をシミュレートできることを示す．

Sim を次のように構成する：$v \xleftarrow{\$} \{0,1\}^\lambda$ とサンプルし，$c = 0$ であるとき，$\mathsf{Sim}(1^\lambda, 0, s_0)$ は $(0, s_0, v)$ を出力し，$c = 1$ のとき，$\mathsf{Sim}(1^\lambda, 1, s_1)$ は $(1, v, s_1)$ を出力する．

ここで，$c = 0$ のとき $\hat{z} = (0, s_0, s_1 + u)$ であり，$c = 1$ のとき $\hat{z} = (1, s_0 + u, s_1)$ である．そして，$u \in \{0,1\}^\lambda$ は一様ランダムにサンプルされているため，$s_0 + u$，$s_1 + u$ と v はそれぞれ同じ $\{0,1\}^\lambda$ の一様分布に従う．よって，$\mathsf{Sim}(1^\lambda, c, f_{\mathrm{SOT}}(c, (s_0, s_1)))$ は $\hat{z}$ を統計的にシミュレート可能である．

上記の議論から，任意の入力 x, y に対して $\mathsf{Sim}(1^\lambda, x, f(x, y)) \approx \hat{z}$ である．よって，上記の構成は OSARE の統計的片側安全性を満たす． $\square$

上記の関数 f_{SOT} に対する OSARE の構成は，セキュリティパラメータ λ に関して，通信複雑性が $O(\lambda)$ である．提案構成に対して，Bitansky らの変換手法を用いることで，最終的には，通信複雑性が $O(\lambda)$ である効率的な任意の関数 f に対する ARE が得られる．

5. まとめと今後の課題

本研究では，SOT の関数 f_{SOT} に対する OSARE 方式を新たに構成し，最終的に，通信複雑性がセキュリティパラメータ λ に関して $O(\lambda)$ である ARE 方式が得られた．既存研究で PKE を用いた ARE の構成が $O(\lambda^2)$ の通信複雑性を要していることから，より効率的な PKE を用いた ARE が実現可能であることを示した．

今後の課題としては，上記の効率化が ARE に対しても同様に可能であるか，つまり，関数 f_{SOT} に対する ARE 方式を直接構成することで，任意の関数 f に対する ARE を効率的に構成可能であるか明らかにすることが挙げられる．加えて，既存の任意の関数 f に対する ARE は GC を用いて構成されているが，GC 以外にも秘密分散法等を用いて MPC を実現できることから，別のアプローチで任意の関数に対する ARE を構成することも今後の課題である．これは，1.2 節で議論したように，GC を用いた ARE の理想的な通信複雑性は $O(\lambda n \ell)$ であるため，ARE の効率化を図るためにも，GC を用いない ARE の実現手法の提案は非常に重要な課題である．

謝辞 本研究は JSPS 科研費 JP23H00468, JP23H00479, JP23K17455, JP23K21644, JP23K24846 JP23KJ0968, の助成，および JST CREST JPMJCR23M2, における AIP チャレンジプログラムの支援を受けたものです．

参考文献

[1] Bitansky, N., Erabelli, S. and Garg, R.: Additive Randomized Encodings from Public Key Encryption (2025).
[2] Blakley, G. R.: Safeguarding cryptographic keys, *Managing requirements knowledge, international workshop on* (1979).
[3] Diffie, W. and Hellman, M. E.: New directions in cryptography, *IEEE* (1976).
[4] Eriguchi, R., Ohara, K., Yamada, S. and Nuida, K.: Non-interactive Secure Multiparty Computation for Symmetric Functions, Revisited: More Efficient Constructions and Extensions, *CRYPTO 2021* (Malkin, T. and Peikert, C., eds.) (2021).
[5] Even, S., Goldreich, O. and Lempel, A.: A randomized protocol for signing contracts, *Communications of the ACM* (1985).
[6] Halevi, S., Ishai, Y., Kushilevitz, E. and Rabin, T.: Additive Randomized Encodings and Their Applications, *CRYPTO 2023* (2023).
[7] Håstad, J., Impagliazzo, R., Levin, L. A. and Luby, M.: A Pseudorandom Generator from any One-way Function, *SIAM* (1999).
[8] Ishai, Y., Kushilevitz, E., Ostrovsky, R. and Sahai, A.: Cryptography from Anonymity, *FOCS2006* (2006).
[9] Rabin, M. O.: How To Exchange Secrets with Oblivious Transfer, *IACR Cryptol. ePrint Arch.* (2005).
[10] Shamir, A.: How to share a secret, *ACM* (1979).
[11] Yao, A. C.: How to Generate and Exchange Secrets (Extended Abstract), *27th Annual Symposium on Foundations of Computer Science* (1986).