ある条件を満たす zero-knowledge arguments of knowledge において健全性を統一的に証明可能なツールの開発

坂根 晋平[1,a)]　篠原 直行[2,b)]　伯田 恵輔[3]

概要：ZKAoKs (Zero-Knowledge Arguments of Knowledge) とは，証明者および検証者の 2 名のパーティ（参加者）が対話を行い，証明者は秘密情報（秘密鍵など）を一切漏らすことなく秘密情報を保持しているという事実のみを検証者に証明するプロトコル（証明者と検証者の組）である．プロトコルが ZKAoK であることを証明するためには，プロトコルが完全性，健全性，およびゼロ知識性と呼ばれる 3 つの性質を満たすことを数学的に証明する必要がある．一方，上記 3 つの性質の数学的な証明はそれぞれのプロトコルに依存しているため統一的に扱う（1 つの定理を用いて複数のプロトコルの性質を証明する）ことは困難である．特に，完全性とゼロ知識性の数学的な証明を統一的に扱うことは困難である．本稿では，上記 3 つの性質のうち，健全性（validity や knowledge soundness と呼ばれる）に焦点を当て，多くの ZKAoKs が満たしている基本的な条件を満足する ZKAoKs の健全性を証明するためのツール（補題）を提案し，本ツールの適用が可能な既存の ZKAoKs の例を挙げる．

キーワード：ゼロ知識証明, zero-knowledge arguments of knowledge, 健全性, validity, knowledge soundness

A tool for systematically proving the soundness of zero-knowledge arguments of knowledge with certain conditions

Shinpei Sakane[1,a)]　Naoyuki Shinohara[2,b)]　Keisuke Hakuta[3]

Abstract: A zero-knowledge argument of knowledge (ZKAoK) is a protocol consisting of two parties called a prover and a verifier. The prover interacts with the verifier and proves knowing its secret information (like a secret key) to the verifier without revealing that secret information. To show a protocol is a ZKAoK, we need to prove that the protocol satisfies 3 properties called completeness, soundness, and zero-knowledge, mathematically. It is hard to show that protocols are ZKAoKs systematically, which means that we prove protocols satisfy the 3 properties with a theorem, because proving the 3 properties (in particular the completeness and the zero-knowledge) relies on each protocol. This paper focuses on the soundness called validity or knowledge soundness. We propose a tool (a lemma) for systematically proving the soundness of ZKAoKs with certain conditions many ZKAoKs satisfy, and provide the examples of how to apply our proposed tool to known ZKAoKs.

Keywords: zero-knowledge proofs, zero-knowledge arguments of knowledge, soundness, validity, knowledge soundness

[1] 島根大学大学院自然科学研究科
Graduate School of Natural Science and Technology, Shimane University
[2] 国立研究開発法人情報通信研究機構サイバーセキュリティ研究所セキュリティ基盤研究室
Security Fundamentals Laboratory, Cybersecurity Institute, National Institute of Information and Communications Technology
[3] 名城大学理工学部数学科
Department of Mathematics, Meijo University
[a)] n21m105@matsu.shimane.ac.jp

1. はじめに

ゼロ知識証明とは，証明者および検証者の 2 名のパーティ（参加者）が対話を行い，証明者は何らかの主張（秘密情報を保持する，など）が成り立つことを検証者に証明するプロトコル（証明者と検証者の組）である．ただし，この対話では，証明者は主張が成り立つこと以外の情報を検証者に一切漏らさない．ゼロ知識証明は，デジタル署名 [1] やブロックチェーン [6] などの多くの実用的な暗号技術として利用されている．

ゼロ知識証明における証明者の主張が「秘密情報（秘密鍵など）を保持する」であるとき，ゼロ知識証明を ZKAoKs (Zero-Knowledge Arguments of Knowledge) という．ZKAoKs は，多くの数学的対象（たとえば，多変数多項式や格子など）を用いて構成することができる．多変数多項式を用いて構成された ZKAoKs として [1]，[7]，[8]，[9] などが知られている．特に，[8] は多変数多項式を用いて構成された代表的な ZKAoKs である．本稿では，ZKAoKs として主に多変数多項式を用いて構成されたものを扱う．

プロトコルが ZKAoK であるためには，プロトコルは以下の 3 つの性質を満たす必要がある [2]:

- 完全性: 秘密情報を保持する証明者が秘密情報を保持していることを検証者に証明できるという性質
- 健全性: 秘密情報を保持しない証明者が秘密情報を保持していることを検証者に証明できないという性質
- ゼロ知識性: 証明者が検証者に秘密情報を一切漏らさないという性質

そのため，プロトコルが ZKAoK であることを証明するためには，プロトコルが上記 3 つの性質を満たすことを数学的に証明する必要がある．プロトコルが ZKAoK であることの証明は個別のプロトコルごとに実施される．これは上記 3 つの性質の数学的な証明はそれぞれのプロトコルに依存しているためである．特に，完全性とゼロ知識性の数学的な証明はそれぞれのプロトコルに大きく依存しており，統一的に扱う（1 つの定理を用いて複数のプロトコルの性質を証明する）ことは困難である．

本稿では，上記 3 つの性質のうち，健全性に焦点を当て，[1]，[4]，[7]，[8]，[9] のように多くの ZKAoKs が満たす基本的な条件を満足する ZKAoKs に対して適用可能な健全性を証明するためのツール（補題）を提案する（第 3 章，補題 3.1）．本ツール（第 3 章，補題 3.1）を用いることにより，多くの既存 ZKAoKs が健全性を満たすことを統一的に証明することができ，さらに今後提案される ZKAoKs の健全性の証明をも与えることができると考えられる．本ツールを用いて健全性を証明することができない ZKAoKs は現時点において見つかっていない．また，本稿では上記ツールを **Extracting Lemma** と命名する（命名理由は第 3 章を参照）．

本稿の構成は以下の通りである．第 2 章では，第 3 章以降で必要な定義，命題，および定理について述べる．第 3 章では，本ツール (Extracting Lemma) の命名理由を説明し，Extracting Lemma の証明を行う．第 4 章では，Extracting Lemma を適用可能な ZKAoKs の例を挙げ，それらが健全性を満たすことを Extracting Lemma を用いて証明する．第 5 章で本稿の結果をまとめる．

2. 準備

ここでは，本稿で用いる記法および定義を述べる．

2.1 記法

$\mathbb{N} := \{1, 2, \ldots\}$ を自然数全体の集合，$\mathbb{R}$ を実数全体の集合，$\mathbb{R}[T]$ を T を変数とする $\mathbb{R}$ 上の 1 変数多項式環，$\mathbb{F}_q$ を元の個数が q の有限体，$\mathbb{F}_q[X_1, \ldots, X_n]$ を $X_1, \ldots, X_n$ を変数とする $\mathbb{F}_q$ 上の n 変数多項式環，λ をセキュリティパラメータ，N を自然数とする．$\sharp S$ で有限集合 S の元の個数を，$|x|$ で x のビット長を，それぞれ表す．$x \in_R S$ を有限集合 S から一様ランダムに x を選ぶことを表す記号，$x \stackrel{?}{=} y$ を x と y が等しいか否かを検証することを表す記号とする．

2.2 二項関係

空でない有限集合 A，B に対し, $R \subseteq A \times B$ を二項関係という．

二項関係 $R \subseteq A \times B$ が次の 2 つの条件を満たすとき，R を NP 関係という（NP 関係の詳細については [2] を参照されたい）:

- 任意の $(x, y) \in R$ に対し，$|y| \leq p(|x|)$ となる多項式 $p \in \mathbb{R}[T]$ が存在する
- 任意の $(x, y) \in A \times B$ に対し，$(x, y) \in R$ か否かを決定するための多項式時間アルゴリズムが存在する

本稿で扱う二項関係はすべて NP 関係とする．

$R \subseteq A \times B$ を NP 関係，$x \in A$ とする．このとき，集合 $R(x)$ および L_R を次のように定める:

$$R(x) := \{y \in B \mid (x, y) \in R\} \subseteq B,$$
$$L_R := \{x \in A \mid \exists y \in B \text{ s.t. } (x, y) \in R\} \subseteq A.$$

$(x, y) \in R$ のとき，y を x に対する解 (solution) と呼ぶ．y が解であることを明示するために，y を s と表すこともある．

2.3 Zero-Knowledge Arguments of Knowledge

ZKAoKs (Zero-Knowledge Arguments of Knowledge) とは，証明者および検証者の 2 名のパーティ（参加者）

b) shnhr@nict.go.jp

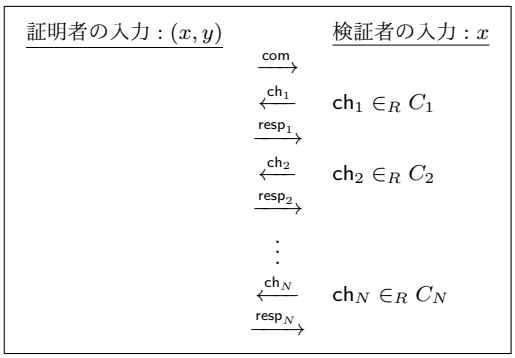

図 1 ZKAoK における証明者と検証者の動作
Fig. 1 A prover and a verifier on a ZKAoK

が対話を行い，証明者は秘密情報（秘密鍵など）を一切漏らすことなく秘密情報を保持しているという事実のみを検証者に証明するプロトコル（証明者と検証者の組）である．証明者は P と，検証者は V と，それぞれ表されることが多い（証明者と検証者の厳密な定義については [2] を参照されたい）．ZKAoKs における証明者と検証者の入力として，以下の 3 つのデータがある:

- 共通入力: 証明者と検証者の両方に与えられる共通のデータ
- 補助入力: 証明者のみに与えられる補助的なデータ
- ランダムな入力: 証明者および検証者が確率的に動作する際に用いるデータ

ZKAoKs では，R を NP 関係，$(x, y) \in R$ とするとき，x を共通入力，y を証明者の補助入力とする．ランダムな入力は一様ランダムに選ばれることが多い．

ZKAoKs における証明者と検証者は図 1 のように動作する．ここで, 図 1 で用いた記号の説明は以下の通りである:

- com: コミットメントと呼ばれるデータ
- ch_i: チャレンジと呼ばれるデータ
- resp_i: レスポンスと呼ばれるデータ
- C_i: 有限集合

したがって，証明者を P と，検証者を V と，それぞれ表すと，ZKAoKs は次のように動作する:

- round 1: P はコミットメント com を計算し，com を V へ送信する
- round 2: V はチャレンジ ch_1 を C_1 から一様ランダムに選び，ch_1 を P へ送信する
- round 3: P は ch_1 に応じてレスポンス resp_1 を計算し，resp_1 を V へ送信する
- round 4: V はチャレンジ ch_2 を C_2 から一様ランダムに選び，ch_2 を P へ送信する
- round 5: P は ch_2 に応じてレスポンス resp_2 を計算し，resp_2 を V へ送信する

  ⋮
- round $2N$: V はチャレンジ ch_N を C_N から一様ランダムに選び，ch_N を P へ送信する
- round $2N+1$: P は ch_N に応じてレスポンス resp_N を計算し，resp_N を V へ送信する

図 1 において，対話（P および V によるデータの送信）が $2N+1$ 回行われているため，図 1 の形のプロトコルを $(2N+1)$-pass プロトコルという．また，P および V が互いに送受信するすべてのデータの組 $(\mathsf{com}, \mathsf{ch}_1, \mathsf{resp}_1, \ldots, \mathsf{ch}_N, \mathsf{resp}_N)$ をトランスクリプトという．

$(2N+1)$-pass プロトコルにおいて，検証者はすべてのレスポンス $\mathsf{resp}_1, \ldots, \mathsf{resp}_N$ を受け取った後（round $2N+1$ の後），共通入力 x およびレスポンス $\mathsf{resp}_1, \ldots, \mathsf{resp}_N$ を用いて，証明者が本当に秘密情報 y を保持するか否かを検証する．したがって，検証者の検証結果は以下の 2 つである:

- 証明者は秘密情報 y を保持する
- 証明者は秘密情報 y を保持しない

検証結果が前者のとき，証明者の主張（秘密情報 y を保持する）は検証者にアクセプトされ，検証結果が後者のとき，証明者の主張は検証者にリジェクトされる．

ZKAoKs は完全性，健全性，およびゼロ知識性と呼ばれる 3 つの性質を満たすプロトコルとして定義される．本稿の主題は ZKAoKs の健全性であり，ZKAoKs の健全性（validity という[*1]）の定義は以下のとおりである．

定義 2.1 (validity)**.** R を NP 関係, κ を写像 $\kappa : \mathbb{N} \to [0, 1]$, V を確率的多項式時間マシンとする．V が validity with error κ であるとは，多項式 $q \in \mathbb{R}[T]$ および確率的オラクルマシン K が存在して，任意の確率的多項式時間マシン P^*, 任意の $x \in L_R$, および任意の $y, r \in \{0,1\}^*$ に対し，K が次を満たすことをいう:

> 共通入力 x, 補助入力 y, そしてランダムな入力 r をもつ証明者 P^* を $P^*_{x,y,r}$ と表す．さらに，入力 x で V が $P^*_{x,y,r}$ をアクセプトする確率を $p(x, y, r)$ と表す．このとき，入力 x およびオラクル $P^*_{x,y,r}$ へのアクセスをもつマシン K が平均的多項式時間で動作し，
>
> $$\Pr[K^{P^*_{x,y,r}}(x) \in R(x)] \geq \frac{p(x, y, r) - \kappa(|x|)}{q(|x|)}$$
>
> が成り立つ．

定義 2.1 におけるオラクルマシン K は x に対する解 y を抽出するマシンである．ZKAoKs の詳細については [2] や [5] などを参照されたい．

2.4 コミットメント方式

コミットメント方式は ZKAoKs などに用いられるプロトコルである．コミットメント方式は次の 2 つのフェーズからなる．

[*1] knowledge soundness とも呼ばれる．

- コミットメントフェーズ: 送信者と呼ばれるパーティが，ある関数（コミットメント関数という）$\mathsf{Com} : \{0,1\}^* \times \{0,1\}^{O(\lambda)} \to \{0,1\}^{O(\lambda)}$ を用いて，$c \leftarrow \mathsf{Com}(s;\rho)$ を計算する．ここで，ρ はランダムに選ばれた文字列を，s は送信者のメッセージを，それぞれ表す．そして，送信者は受信者と呼ばれるパーティへ c を送信する
- 開示フェーズ: 送信者は (s,ρ) を受信者へ与え，受信者は $c \stackrel{?}{=} \mathsf{Com}(s;\rho)$ を検証する

本稿では，computationally binding という性質をもつコミットメント関数を用いる．computationally binding とは, コミットメントフェーズの後，送信者が $\mathsf{Com}(s;\rho) = \mathsf{Com}(s';\rho')$ なる $s' \neq s$ および ρ' を効率的に生成できないことを意味する．このようなコミットメント関数の構成方法については [3] を参照されたい．

2.5 有限体上の多変数多項式系

有限体 $\mathbb{F}_q$ 上の多変数多項式系 とは，$\mathbb{F}_q$ 上の多変数多項式の組，すなわち，$\boldsymbol{F} = (f_1,\ldots,f_m) \in \mathbb{F}_q[X_1,\ldots,X_n]^m$ のことをいう．以下 $\boldsymbol{X} := (X_1,\ldots,X_n)$, $\boldsymbol{Y} := (Y_1,\ldots,Y_n)$ とする．

有限体上の多変数多項式系 $\boldsymbol{F} = (f_1,\ldots,f_m)$ でかつ，すべての $1 \leq \ell \leq m$ に対し，$\deg f_\ell = 2$ および $f_\ell(\boldsymbol{0}) = 0$ が成り立つもの全体の集合を $\mathcal{MQ}(n,m,\mathbb{F}_q)$ とする．すなわち，任意の $\boldsymbol{F} = (f_1,\ldots,f_m) \in \mathcal{MQ}(n,m,\mathbb{F}_q)$ における各多項式 f_ℓ $(1 \leq \ell \leq m)$ は次の形をしている:

$$f_\ell(\boldsymbol{X}) = \sum_{1\leq i\leq j\leq n} a_{\ell,i,j}X_iX_j + \sum_{1\leq i\leq n} b_{\ell,i}X_i, \quad a_{\ell,i,j}, b_{\ell,i} \in \mathbb{F}_q.$$

次に，多変数 2 次多項式系に関する二項関係 R_2 を次のように定義する．任意の $\boldsymbol{F} \in \mathcal{MQ}(n,m,\mathbb{F}_q)$ に対し，

$$R_2 := \{((\boldsymbol{F},\boldsymbol{v}),\boldsymbol{s}) \mid \boldsymbol{v} = \boldsymbol{F}(\boldsymbol{s})\} \subseteq (\{\boldsymbol{F}\} \times \mathbb{F}_q^m) \times \mathbb{F}_q^n.$$

このとき，R_2 は NP 関係である．

次に，多変数 2 次多項式系に基づく ZKAoKs [1], [8] に用いられる双線形写像 $\boldsymbol{G} : \mathbb{F}_q^n \times \mathbb{F}_q^n \to \mathbb{F}_q^m$ を定義する．任意の $\boldsymbol{F} = (f_1,\ldots,f_m) \in \mathcal{MQ}(n,m,\mathbb{F}_q)$ に対し，写像 $\boldsymbol{G} = (g_1,\ldots,g_m)$ を次のように定義する:

$$\boldsymbol{G}(\boldsymbol{X},\boldsymbol{Y}) := \boldsymbol{F}(\boldsymbol{X}+\boldsymbol{Y}) - \boldsymbol{F}(\boldsymbol{X}) - \boldsymbol{F}(\boldsymbol{Y}).$$

このとき，$\boldsymbol{G} = (g_1,\ldots,g_m)$ の各成分 g_ℓ $(1 \leq \ell \leq m)$ を明示的に表すと次のようになる:

$$g_\ell(\boldsymbol{X},\boldsymbol{Y}) = \sum_{1\leq i\leq j\leq n} a_{\ell,i,j}(X_iY_j + Y_iX_j). \qquad (2.1)$$

ここで，$a_{\ell,i,j}$ は f_ℓ における次数 2 の単項式の係数である．$\boldsymbol{G}$ が双線形写像であることは式 (2.1) より明らかである．

3. Extracting Lemma: ある条件を満たす zero-knowledge arguments of knowledge において健全性を統一的に証明可能なツール

本章では，多くの ZKAoKs が満たしている基本的な条件を満足する ZKAoKs に対して適用可能な健全性 (validity) を証明するためのツール（補題）を提案する．本ツールの主要部分は解の「抽出 (extraction)」であるため，本ツールを Extracting Lemma と命名する．Extracting Lemma を補題 3.1 に示す．

補題 3.1 (Extracting Lemma)．R を NP 関係，P および V を確率的多項式時間マシン，(P,V) を $(2N+1)$-pass プロトコル，$C_1,\ldots,C_N$ を $\sharp C_1,\ldots,\sharp C_N \geq 2$ なる有限集合，$\kappa : \mathbb{N} \to [2^{-N},1]$, $(x,y) \in R$ とし，さらに

$$\mathcal{T} := \left\lfloor \kappa(|x|)\left(\prod_{i=1}^{N} \sharp C_i\right)\right\rfloor + 1$$

と定める．ただし，$\mathcal{T}$ は $|x|$ の多項式で上から抑えられるとする．さらに，$1 \leq k \leq \mathcal{T}$ を固定し，f を多項式時間で計算可能な写像 $f : C_1^k \times C_2^k \times \cdots \times C_N^k \to \{0,1\}$ とする．このとき，次を満たす多項式時間アルゴリズム A が存在すると仮定する: ある t 個のアクセプトされるトランスクリプト

$$(\mathsf{com},\mathsf{ch}_1^{(1)},\ldots,\mathsf{resp}_N^{(1)}),\ldots,(\mathsf{com},\mathsf{ch}_1^{(k)},\ldots,\mathsf{resp}_N^{(k)})$$

に対し，

$$f(\mathsf{ch}_1^{(1)},\ldots,\mathsf{ch}_1^{(k)},\ldots,\mathsf{ch}_N^{(1)},\ldots\mathsf{ch}_N^{(k)}) = 1$$

が成り立つならば，A は x およびこれらの値を入力としてとり，x の解を確率 1 で出力する．また，任意の証明者 P^* に対し，集合 $\mathcal{C}$ を $P^*_{x,y,r}$ と V が対話してアクセプトされるチャレンジの組の集合とする．さらに，もし $\sharp\mathcal{C} \geq \mathcal{T}$ ならば，$\mathcal{C}$ の任意の $\mathcal{T}$ 個の元

$$(\mathsf{ch}_1^{(1)},\ldots,\mathsf{ch}_N^{(1)}),\ldots,(\mathsf{ch}_1^{(\mathcal{T})},\ldots,\mathsf{ch}_N^{(\mathcal{T})}) \in \mathcal{C}$$

の中に（これらを適切に並べ替えると），

$$f(\mathsf{ch}_1^{(1)},\ldots,\mathsf{ch}_1^{(k)},\ldots,\mathsf{ch}_N^{(1)},\ldots\mathsf{ch}_N^{(k)}) = 1$$

を満たす

$$(\mathsf{ch}_1^{(1)},\ldots,\mathsf{ch}_N^{(1)}),\ldots,(\mathsf{ch}_1^{(k)},\ldots,\mathsf{ch}_N^{(k)})$$

が存在すると仮定する．このとき，V は validity with error κ を満たす．

補題 3.1 を証明するためには，補題 3.2 および補題 3.3 が必要である．まず，補題 3.2 を証明する．

補題 3.2. $n \in \mathbb{N}$ とする．$a,b \in \{0,1\}^n$ に対し，a と b が

等しいか否か判定するために必要な時間計算量は $O(n)$ である．

Proof. a, b をそれぞれビット列で表し，

$$a = a_{n-1} \cdots a_0, \quad b = b_{n-1} \cdots b_0,$$

$a_0, \ldots, a_{n-1}, b_0, \ldots, b_{n-1} \in \{0,1\}$ とおく．このとき，a と b が等しいか否か判定するためには，すべての $0 \leq i \leq n-1$ に対し，2 つのビット a_i および b_i が等しいか否か判定すればよい．したがって，a と b が等しいか否か判定するためには，2 つのビットが等しいか否かを高々 n 回比較すればよいため，必要な時間計算量は $O(n)$ である． □

次に，補題 3.3 を証明する．

補題 3.3. $x \in \{0,1\}^*$ とする．$\mathcal{T} \in \mathbb{N}$ を $|x|$ の多項式で上から抑えることができると仮定する．このとき，$1 \leq k \leq \mathcal{T}$ に対し $\binom{\mathcal{T}}{k}$ も $|x|$ の多項式で上から抑えることができる．

Proof. $\binom{\mathcal{T}}{k} \leq \mathcal{T}^k$ が成り立つ．実際，

$$\begin{aligned}\binom{\mathcal{T}}{k} &= \frac{(\mathcal{T}+1)!}{(\mathcal{T}-k)!k!} = \frac{\mathcal{T}(\mathcal{T}-1)\cdots(\mathcal{T}-(k-1))}{k!} \\ &\leq \mathcal{T}(\mathcal{T}-1)\cdots(\mathcal{T}-(k-1)) \leq \mathcal{T}^k.\end{aligned}$$

したがって，$\mathcal{T}$ は $|x|$ の多項式で上から抑えることができるため，$\binom{\mathcal{T}}{k}$ も $|x|$ の多項式で上から抑えることができる． □

補題 3.2 および補題 3.3 を用いて補題 3.1 を示す．

Proof of 補題 3.1. 任意の証明者 P^*, $x \in L_R$, $y, r \in \{0,1\}^*$ をそれぞれ固定する．

本証明は次の 2 つの段階からなる．まず，入力 x およびオラクル $P^*_{x,y,r}$ へのアクセスをもつ確率的オラクルマシン K を構成し，K の平均実行時間が $|x|$ の多項式であることを示す．次に，K が x の解を出力する確率を $\Pr_K$ とするとき，$\Pr_K \geq p(x,y,r) - \kappa(|x|)$ が成り立つことを証明する．すなわち，

$$\Pr_K := \Pr[K^{P^*_{x,y,r}}(x) \in R(x)] \geq p(x,y,r) - \kappa(|x|) \tag{3.1}$$

が成り立つことを証明する．式 (3.1) を示すことにより，V が validity with error κ を満たすことを証明できる（定義 2.1 における多項式 $q \in \mathbb{R}[T]$ として 1 をとればよい）．

x およびオラクル $P^*_{x,y,r}$ へのアクセスをもつ確率的オラクルマシン K を次のように構成する．

step 1. K は検証者 V を演じ，$P^*_{x,y,r}$ と対話し，トランスクリプト

$$\mathrm{tr}^{(1)} = (\mathsf{com}, \mathsf{ch}_1^{(1)}, \ldots, \mathsf{resp}_N^{(1)})$$

を生成する．$\mathrm{tr}^{(1)}$ がアクセプトされないならば，$\perp$ を出力し，停止する．$\mathrm{tr}^{(1)}$ がアクセプトされるならば，step 2 に進む．

step 2. $2 \leq i \leq \mathcal{T}$ に対し，以下を独立に繰り返す（step 1 とも独立に行う）：

step 2-1. $P^*_{x,y,r}$ を最初の状態に巻き戻す（$P^*_{x,y,r}$ はランダムな入力 r を固定しているため，$P^*_{x,y,r}$ が V に送信する最初のメッセージ com は不変であることに注意されたい）．

step 2-2. K は検証者 V を演じ，$P^*_{x,y,r}$ と対話し，トランスクリプト

$$\mathrm{tr}^{(i)} = (\mathsf{com}, \mathsf{ch}_1^{(i)}, \ldots, \mathsf{resp}_N^{(i)})$$

を生成する．

step 2-1 および step 2-2 を $\mathrm{tr}^{(i)}$ がアクセプトされるトランスクリプトになるまで繰り返す．アクセプトされるトランスクリプト $\mathrm{tr}^{(i)}$ を生成できた場合，i をインクリメントし，再度 step 2 を実行する．ただし，i をインクリメントした結果，$i = \mathcal{T}+1$ となるならば step 3 に進む．

step 3. 集合 C を K が生成した $\mathcal{T}$ 個のトランスクリプトにおけるチャレンジのみを集めた集合とする．すなわち，集合 C を次のように定める:

$$C := \{(\mathsf{ch}_1^{(1)}, \ldots, \mathsf{ch}_N^{(1)}), \ldots, (\mathsf{ch}_1^{(T)}, \ldots, \mathsf{ch}_N^{(\mathcal{T})})\}.$$

このとき，$\sharp C < \mathcal{T}$ （すなわち，ある $1 \leq i < j \leq \mathcal{T}$ が存在し $(\mathsf{ch}_1^{(i)}, \ldots, \mathsf{ch}_N^{(i)}) = (\mathsf{ch}_1^{(j)}, \ldots, \mathsf{ch}_N^{(j)})$ となる）ならば，$\perp$ を出力する．そうでない ($\sharp C = \mathcal{T}$ が成り立つ) ならば，トランスクリプト $\mathrm{tr}^{(1)}, \ldots, \mathrm{tr}^{(\mathcal{T})}$ に対し，（必要ならば順番を入れ替えて）

$$f(\mathsf{ch}_1^{(1)}, \ldots, \mathsf{ch}_1^{(k)}, \ldots, \mathsf{ch}_N^{(1)}, \ldots, \mathsf{ch}_N^{(k)}) = 1 \quad (3.2)$$

を満たす $\mathrm{tr}^{(1)}, \ldots, \mathrm{tr}^{(k)}$ が存在するか否かを判定する．式 (3.2) を満たす $\mathrm{tr}^{(1)}, \ldots, \mathrm{tr}^{(k)}$ が存在するならば，A を次のように実行し，s を出力し，停止する:

$$s \leftarrow A(\mathsf{com}, \mathsf{ch}_1^{(1)}, \ldots, \mathsf{resp}_N^{(1)}, \ldots, \mathsf{ch}_1^{(k)}, \ldots, \mathsf{resp}_N^{(k)}).$$

式 (3.2) を満たす $\mathrm{tr}^{(1)}, \ldots, \mathrm{tr}^{(k)}$ が存在しないならば，$\perp$ を出力し，停止する．

記法を簡単にするために，$1 \leq i \leq \mathcal{T}$ に対し，$\mathsf{ch}^{(i)}$ を次のように定める:

$$\mathsf{ch}^{(i)} := (\mathsf{ch}_1^{(i)}, \ldots, \mathsf{ch}_N^{(i)}).$$

K の平均実行時間が $|x|$ の多項式であることを示す．K の step 1 の実行時間は $|x|$ の多項式である．なぜならば，$P^*_{x,y,r}$ はオラクルであり，K は（確率的多項式時間で動作する）検証者 V を演じるためである．そこで，step 1 の実行時間を $q_1(|x|)$ $(q_1 \in \mathbb{R}[T])$ とおく．

K の step 2 の実行時間を求める．まず，ある $2 \le i \le \mathcal{T}$ に対し，K の step 2 の実行時間を求める．step 2-1 では，$P^*_{x,y,r}$ を初期状態に戻すだけであるため，step 2-1 の実行時間は $|x|$ の多項式である．step 2-2 において，$\mathrm{tr}^{(i)}$ を生成するために必要な実行時間は step 1 と同様の理由により，$|x|$ の多項式である．ゆえに，step 2-1 および step 2-2 をそれぞれ 1 回実行する際に必要な時間は $|x|$ の多項式であるため，この実行時間を $q_{2,i}(|x|)$ $(q_{2,i} \in \mathbb{R}[T],\ 2 \le i \le \mathcal{T})$ とおく．また，step 2 では，アクセプトされる $\mathrm{tr}^{(i)}$ を生成できるまで繰り返し step 2-1 および step 2-2 を実行する．$P^*_{x,y,r}$ がアクセプトされる確率は $p(x,y,r)$ であるため，この繰り返しの回数の平均は $1/p(x,y,r)$ である．したがって，step 2 の各 $2 \le i \le \mathcal{T}$ の実行時間は $q_{2,i}(|x|)/p(x,y,r)$ である．ゆえに，step 2 の実行時間として次を得る:

$$\frac{1}{p(x,y,r)}(q_{2,2}(|x|) + \cdots + q_{2,T}(|x|)).$$

したがって，

$$q_2(|x|) := q_{2,2}(|x|) + \cdots + q_{2,T}(|x|)$$

とおくと，step 2 の実行時間は $q_2(|x|)/p(x,y,r)$ である．

最後に，K の step 3 の実行時間を求める．step 3 では，まず $\sharp C = \mathcal{T}$ であるか否かを判定する．これは $1 \le i < j \le \mathcal{T}$ に対し，$\mathsf{ch}^{(i)} = \mathsf{ch}^{(j)}$ が成り立つか否かを判定すればよい．これを判定するためには，各 $1 \le \ell \le N$ に対し，$\mathsf{ch}^{(i)}_\ell$ と $\mathsf{ch}^{(j)}_\ell$ が等しいか否かを判定すればよい．これは補題 3.2 により，$O(\lceil \log_2 \sharp C_\ell \rceil)$ で実行可能である．ゆえに，この判定に必要な時間計算量は $O(\lceil \log_2 \sharp C_1 \rceil) + \cdots + O(\lceil \log_2 \sharp C_N \rceil)$ である．仮定により，$\mathcal{T}$ は $|x|$ の多項式で上から抑えることができるため，$\lceil \log_2 \sharp C_1 \rceil, \ldots, \lceil \log_2 \sharp C_N \rceil$ も $|x|$ の多項式で上から抑えることができる．そのため，この比較に必要な時間計算量 $O(\lceil \log_2 \sharp C_1 \rceil) + \cdots + O(\lceil \log_2 \sharp C_N \rceil)$ も $|x|$ の多項式で上から抑えることができる．さらに，この比較は最大 $\binom{\mathcal{T}}{2}$ 回行われ，$\mathcal{T}$ が $|x|$ の多項式で上から抑えることができるため，補題 3.3 により，$\binom{\mathcal{T}}{2}$ も $|x|$ の多項式で上から抑えることができる．したがって，$\sharp C = \mathcal{T}$ であるか否かの判定に必要な時間計算量は $|x|$ の多項式である．

次に，step 3 における $\sharp C = \mathcal{T}$ の場合における残りの実行時間を求める．まず，式 (3.2) を満たす $\mathrm{tr}^{(1)}, \ldots \mathrm{tr}^{(k)}$ が存在するか否かを判定する箇所の実行時間を考える．f の計算は仮定により多項式時間で可能であるため，$\mathrm{tr}^{(1)}, \ldots, \mathrm{tr}^{(\mathcal{T})}$ から $k \le \mathcal{T}$ 個を取り出す回数が多項式であることを示せばよい．$\mathrm{tr}^{(1)}, \ldots, \mathrm{tr}^{(\mathcal{T})}$ の中から k 個を取り出す回数は $\binom{\mathcal{T}}{k}$ である．$\mathcal{T}$ は仮定により $|x|$ の多項式で上から抑えることができるため，補題 3.3 により，$\binom{\mathcal{T}}{k}$ も $|x|$ の多項式で上から抑えることができる．最後に実行する A は多項式時間アルゴリズムであるため，A の実行も $|x|$ の多項式で可能である．したがって，step 3 の実行時間は $|x|$ の多項式であるため，step 3 の実行時間を $q_3(|x|)$ $(q_3 \in \mathbb{R}[T])$ とおく．

上記の議論により，K の平均実行時間は次の式で上から抑えることができる:

$$\begin{aligned}
&(1 - p(x,y,r))q_1(|x|) \\
&\quad + 3p(x,y,r)\left\{q_1(|x|) + \frac{1}{p(x,y,r)}q_2(|x|) + q_3(|x|)\right\} \\
&= (2p(x,y,r)+1)q_1(|x|) + 3q_2(|x|) + 3p(x,y,r)q_3(|x|).
\end{aligned}$$

したがって，K の平均実行時間は $|x|$ の多項式で上から抑えることができる．

次に，式 (3.1) が成り立つことを示す．式 (3.1) が成り立つことの証明を $p(x,y,r) \le \kappa(|x|)$ の場合，および $p(x,y,r) > \kappa(|x|)$ の場合に分けて示す．

Case 1. $p(x,y,r) \le \kappa(|x|)$: 式 (3.1) が成り立つことは明らかである．

Case 2. $p(x,y,r) > \kappa(|x|)$: $\mathcal{C}$ を $P^*_{x,y,r}$ と V が対話してアクセプトされるチャレンジの組の集合とする（本補題の主張を参照）．このとき，仮定により $\sharp\mathcal{C}$ を次のように評価することができる:

$$\sharp\mathcal{C} = p(x,y,r)\prod_{i=1}^{N}\sharp C_i > \kappa(|x|)\prod_{i=1}^{N}\sharp C_i.$$

また，$\left\lfloor \kappa(|x|)\prod_{i=1}^{N}\sharp C_i \right\rfloor + 1$ は $\kappa(|x|)\prod_{i=1}^{N}\sharp C_i$ より大きい最小の整数であるため

$$\sharp\mathcal{C} \ge \left\lfloor \kappa(|x|)\prod_{i=1}^{N}\sharp C_i \right\rfloor + 1 = \mathcal{T}$$

が成り立つ．ゆえに，$C \subseteq \mathcal{C}$ であるため，K が step 3 に到達し，$\sharp C = \mathcal{T}$ が成り立つならば，仮定により，C の $\mathcal{T}$ 個の元 $\mathsf{ch}^{(1)}, \ldots, \mathsf{ch}^{(\mathcal{T})} \in C$ を適切に並べ替えることにより，

$$f(\mathsf{ch}^{(1)}_1, \ldots, \mathsf{ch}^{(k)}_1, \ldots, \mathsf{ch}^{(1)}_N, \ldots, \mathsf{ch}^{(k)}_N) = 1$$

を満たす $\mathsf{ch}^{(1)}, \ldots, \mathsf{ch}^{(k)} \in C$ が存在する．よって K が step 3 に到達し，$\sharp C = \mathcal{T}$ が成り立つならば，K は確率 1 で s を出力する．したがって，K が s を出力する確率は step 1 および step 3 で決まる．そこで，step 1 および step 3 に着目する．

確率評価を簡単にするために，K が $\perp$ を出力する確率を求める．K が $\perp$ を出力するのは次の 2 つの事象のいずれかが成り立つときである．

- step 1 において，生成したトランスクリプト $\mathrm{tr}^{(1)}$ がアクセプトされない事象．本事象を $\neg\mathrm{acc}_1$ とおく
- step 1 において，生成したトランスクリプト $\mathrm{tr}^{(1)}$ がアクセプトされ，step 3 において，$\sharp C < \mathcal{T}$ となる事象．すなわち，$\mathrm{tr}^{(1)}$ がアクセプトされ，ある $1 \le i < j \le \mathcal{T}$ が存在し

$$(\mathsf{ch}^{(i)}_1, \ldots, \mathsf{ch}^{(i)}_N) = (\mathsf{ch}^{(j)}_1, \ldots, \mathsf{ch}^{(j)}_N)$$

となる事象．本事象を $\mathrm{acc}_1 \wedge \mathrm{col}_{ij}$ とおく

2 つの事象 $\neg\mathrm{acc}_1$ および $\mathrm{acc}_1 \wedge \mathrm{col}_{ij}$ の確率を求める．

まず，$\Pr[\neg\mathrm{acc}_1] = 1 - \Pr[\mathrm{acc}_1]$ であるため，$\Pr[\mathrm{acc}_1]$ を求める．$\Pr[\mathrm{acc}_1]$ は $p(x,y,r)$ の定義により，$\Pr[\mathrm{acc}_1] = p(x,y,r)$ である．ゆえに $\Pr[\neg\mathrm{acc}_1] = 1 - p(x,y,r)$ である．

次に，$\Pr[\mathrm{acc}_1 \wedge \mathrm{col}_{ij}]$ を求める．$\Pr[\mathrm{acc}_1 \wedge \mathrm{col}_{ij}]$ は次のように表すことができる:

$$\begin{aligned}\Pr[\mathrm{acc}_1 \wedge \mathrm{col}_{ij}] &= \Pr[\mathrm{col}_{ij} \wedge \mathrm{acc}_1] \\ &= \Pr[\mathrm{col}_{ij} \mid \mathrm{acc}_1]\Pr[\mathrm{acc}_1] \\ &= \Pr[\mathrm{col}_{ij} \mid \mathrm{acc}_1]p(x,y,r).\end{aligned}$$

ゆえに $\Pr[\mathrm{col}_{ij} \mid \mathrm{acc}_1]$ を求めればよい．$\Pr[\mathrm{col}_{ij} \mid \mathrm{acc}_1]$ を $i = 1$ の場合，および $2 \le i$ の場合に分けて求める．記法を簡単にするために $S := C_1 \times \cdots \times C_N$ とする．

Case A. $i = 1$:

$$\begin{aligned}\Pr[\mathrm{col}_{1j} \mid \mathrm{acc}_1] &= \Pr_{\mathsf{ch}^{(1)} \in S, \mathsf{ch}^{(j)} \in C}[\mathsf{ch}^{(1)} = \mathsf{ch}^{(j)} \mid \mathrm{acc}_1] \\ &= \Pr_{\mathsf{ch}^{(1)}, \mathsf{ch}^{(j)} \in C}[\mathsf{ch}^{(1)} = \mathsf{ch}^{(j)}] \\ &= \sum_{a \in S} \Pr_{\mathsf{ch}^{(1)}, \mathsf{ch}^{(j)} \in C}[\mathsf{ch}^{(1)} = a \wedge \mathsf{ch}^{(j)} = a] \\ &= \sum_{a \in S} \Pr_{\mathsf{ch}^{(1)} \in C}[\mathsf{ch}^{(1)} = a] \Pr_{\mathsf{ch}^{(j)} \in C}[\mathsf{ch}^{(j)} = a] \\ &= \sum_{a \in S} \left(\Pr_{\mathsf{ch} \in C}[\mathsf{ch} = a]\right)^2 \\ &= \sum_{a \in C} \left(\Pr_{\mathsf{ch} \in C}[\mathsf{ch} = a]\right)^2 = \sum_{a \in C} (\sharp C)^{-2} \\ &= (\sharp C)^{-1} = \left(p(x,y,r)\prod_{\ell=1}^{N} \sharp C_\ell\right)^{-1}.\end{aligned}$$

Case B. $i \ge 2$: $i = 1$ の場合と同様に計算を行うことにより，次を得る:

$$\Pr[\mathrm{col}_{ij} \mid \mathrm{acc}_1] = \left(p(x,y,r)\prod_{\ell=1}^{N} \sharp C_\ell\right)^{-1}.$$

ゆえに，$i = 1$ および $i \ge 2$ のどちらの場合においても

$$\Pr[\mathrm{col}_{ij} \mid \mathrm{acc}_1] = \left(p(x,y,r)\prod_{\ell=1}^{N} \sharp C_\ell\right)^{-1}$$

が成り立つ．よって次を得る:

$$\begin{aligned}\Pr[\mathrm{col}_{ij} \wedge \mathrm{acc}_1] &= \Pr[\mathrm{col}_{ij} \mid \mathrm{acc}_1]p(x,y,r) \\ &= \left(p(x,y,r)\prod_{\ell=1}^{N} \sharp C_\ell\right)^{-1} p(x,y,r) \\ &= \left(\prod_{\ell=1}^{N} \sharp C_\ell\right)^{-1}.\end{aligned}$$

したがって，K が $\perp$ を出力する確率として，次を得る:

$$\Pr[\neg\mathrm{acc}_1] + \Pr[\mathrm{col}_{ij} \wedge \mathrm{acc}_1] = 1 - p(x,y,r) + \left(\prod_{\ell=1}^{N} \sharp C_\ell\right)^{-1}. \tag{3.3}$$

最後に，式 (3.1) が成り立つことを示す．式 (3.3) により，K が s を出力する確率として，次を得る:

$$\begin{aligned}\mathrm{Pr}_K &= 1 - (\Pr[\neg\mathrm{acc}_1] + \Pr[\mathrm{col}_{ij} \wedge \mathrm{acc}_1]) \\ &= 1 - \left\{1 - p(x,y,r) + \left(\prod_{\ell=1}^{N} \sharp C_\ell\right)^{-1}\right\} \\ &= p(x,y,r) - \left(\prod_{\ell=1}^{N} \sharp C_\ell\right)^{-1}. \end{aligned} \tag{3.4}$$

また，$\sharp C_1, \ldots, \sharp C_N \ge 2$ であるため，

$$\left(\prod_{\ell=1}^{N} \sharp C_\ell\right)^{-1} \le 2^{-N} \tag{3.5}$$

を得る．式 (3.4), 式 (3.5), および $\kappa(|x|) \ge 2^{-N}$ により，次が成り立つ:

$$\begin{aligned}&\mathrm{Pr}_K - (p(x,y,r) - \kappa(|x|)) \\ &= \mathrm{Pr}_K - p(x,y,r) + \kappa(|x|) \\ &= p(x,y,r) - \left(\prod_{\ell=1}^{N} \sharp C_\ell\right)^{-1} - p(x,y,r) + \kappa(|x|) \\ &= \kappa(|x|) - \left(\prod_{\ell=1}^{N} \sharp C_\ell\right)^{-1} \ge \kappa(|x|) - 2^{-N} \\ &\ge 2^{-N} - 2^{-N} = 0.\end{aligned}$$

よって，式 (3.1) が成り立つため，V は validity with error κ を満たす． □

4. Extracting Lemma の適用例

本章では，既存 ZKAoKs に Extracting Lemma（補題 3.1）を適用し，既存 ZKAoKs が健全性 (validity) を満たすことの別証明を与える．第 1 章でも述べたように，既存 ZKAoKs として，主に多変数多項式に基づく ZKAoKs を挙げ，これらに Extracting Lemma を適用する．ただし，多変数多項式に基づく ZKAoKs 以外にも，Extracting Lemma を適用可能である．たとえば，格子に基づく既存 ZKAoK [4] に対しても Extracting Lemma を適用することができる．本ツールを用いて健全性を証明することができない ZKAoKs は現時点において見つかっていない．

Extracting Lemma の適用が可能な多変数多項式に基づく既存 ZKAoKs として [1], [7], [8], [9] などがある．これらのプロトコルへの Extracting Lemma の適用方法に大きな違いはないため，本節では，代表的な変数多項式に基づく既存 ZKAoK である作本らの 3-pass プロトコル [8] にのみ Extracting Lemma を適用し，作本らの 3-pass プロ

トコルが健全性を満たすことの別証明を与える．

以下，作本らの 3-pass プロトコルが健全性を満たすことを Extracting Lemma を用いて証明する．すなわち，下記の命題 4.1 が成り立つことを Extracting Lemma を用いて証明する．

命題 4.1. Com が computationally binding であると仮定する．このとき，作本らの 3-pass プロトコルは validity with error 2/3 を満たす．

Proof. 作本らの 3-pass プロトコルが Extracting Lemma の条件を満たすことを示す．まず，R_2 は NP 関係である．そして，本プロトコルは 3-pass プロトコルであるため，Extracting Lemma で用いた記号 N は $N = 1$ である．作本らの 3-pass プロトコルの検証者のチャレンジの集合を $C_1 = \{0, 1, 2\}$ とすると，$\sharp C_1 \geq 2$ が成り立つ．また，κ は定値写像であるため，$\kappa(\mathbb{N}) = \{2/3\} \subseteq [2^{-1}, 1]$ が成り立つ．したがって，κ は写像 $\kappa : \mathbb{N} \to [2^{-1}, 1]$ である．

$x \in L_{R_2}$ に対し，$\mathcal{T}$ は次のようになる:

$$\mathcal{T} = \lfloor \kappa(|x|) \times \sharp C_1 \rfloor + 1 = \left\lfloor \frac{2}{3} \times 3 \right\rfloor + 1 = 3 \leq |x|.$$

ここで，以下が成り立つことを用いた:

$$|x| = m \left\{ \frac{(n+1)n}{2} + n + 1 \right\} \lceil \log_2 q \rceil \geq 3.$$

したがって，$\mathcal{T}$ は $|x|$ の多項式で上から抑えることができる．

$k = \mathcal{T} = 3$ とし，写像 $f : C_1^k \to \{0, 1\}$ を次のように定義する:

$$(\mathsf{ch}^{(1)}, \mathsf{ch}^{(2)}, \mathsf{ch}^{(3)}) \mapsto \begin{cases} 1, & \{\mathsf{ch}^{(1)}, \mathsf{ch}^{(2)}, \mathsf{ch}^{(3)}\} = C_1, \\ 0, & \text{otherwise.} \end{cases}$$

このとき，f は多項式時間で計算可能である．

アルゴリズム A を次のように構成する．A は 3 つのトランスクリプト $(\mathsf{com}, \mathsf{ch}^{(1)}, \mathsf{resp}^{(1)})$, $(\mathsf{com}, \mathsf{ch}^{(2)}, \mathsf{resp}^{(2)})$, $(\mathsf{com}, \mathsf{ch}^{(3)}, \mathsf{resp}^{(3)})$ に対し，

$$x, \mathsf{com}, \mathsf{ch}^{(1)}, \mathsf{resp}^{(1)}, \mathsf{ch}^{(2)}, \mathsf{resp}^{(2)}, \mathsf{ch}^{(3)}, \mathsf{resp}^{(3)}$$

を入力としてとり，$\mathsf{resp}^{(1)}, \mathsf{resp}^{(2)}, \mathsf{resp}^{(3)}$ に $\boldsymbol{r}_0, \boldsymbol{r}_1$ が存在すれば，$\boldsymbol{r}_0 + \boldsymbol{r}_1$ を出力する（$\boldsymbol{r}_0$ および $\boldsymbol{r}_1$ については [8] を参照されたい）．このように A を構成すると，A は多項式時間アルゴリズムであり，もし A に入力された 3 つのチャレンジが f で写して 1 となるならば，A は確率 1 で $x \in L_{R_2}$ に対する解 $s \in R_2(x)$ を出力する（詳細については [8] を参照されたい）．

集合 $\mathcal{C} \subseteq C_1$ を Extracting Lemma で定義したものとする．$\sharp\mathcal{C} \geq \mathcal{T} = 3$ ならば，$\mathcal{C} \subseteq C_1 = \{0, 1, 2\}$ より $\mathcal{C} = C_1$ となる．ゆえに，$\mathcal{C}$ の任意の $\mathcal{T} = k = 3$ 個の元 $\mathsf{ch}^{(1)}, \mathsf{ch}^{(2)}, \mathsf{ch}^{(3)}$ は $f(\mathsf{ch}^{(1)}, \mathsf{ch}^{(2)}, \mathsf{ch}^{(3)}) = 1$ を満たす．

よって，作本らの 3-pass プロトコルは Extracting Lemma の条件を満たすため，主張が成り立つ． □

5. まとめ

本稿では，多くの ZKAoKs が満たしている基本的な条件を満足する ZKAoKs の健全性を証明するためのツール（補題）である Extracting Lemma （補題 3.1）を提案した．本ツールを用いることにより，多くの既存 ZKAoKs が健全性を満たすことを統一的に証明することが可能となった．実際，多変数多項式に基づく代表的な ZKAoK [8] に本ツールを適用し，この ZKAoK に健全性が成り立つことを統一的に証明可能であることを示した．これ以外にも本ツールを適用可能な例として [1], [4], [7], [9] などがある．さらに，本ツールは今後提案される他の ZKAoKs の健全性の証明をも与えることができると考えられる．今後の課題として，本ツールを適用することができない ZKAoKs を探すことなどが挙げられる．

参考文献

[1] H. Furue, D. Hoang Duong, and T. Takagi, An Efficient $\mathcal{MQ}$-based Signature with Tight Security Proof, International Journal of Networking and Conputing, vol. 10, no. 2, 308–324, 2020.

[2] O. Goldreich, Foundations of Cryptography: Volume 1, Basic Tools, Cambridge University Press, 2004.

[3] S. Halevi, S. Micali, Practical and Provably-Secure Commitment Schemes from Collision-Free Hashing, In: N. Koblitz (eds.), Advances in Cryptology - CRYPT 1996, LNCS, vol. 1109, 201–215, Springer, Berlin, Heidelberg (1996).

[4] A. Kawachi, K. Tanaka, and K. Xagawa, Concurrently Secure Identification Schemes Based on the Worst-Case Hardness of Lattice Problems, In: J. Pieprzyk (eds), Advances in Cryptology - ASIACRYPT 2008, LNCS, vol. 5350, 372–389, Springer, Berlin, Heidelberg (2008).

[5] Yehuda Lindell, Parallel Coin-Tossing and Constant-Round Secure Two-Party Computation, Journal of Cryptology, vol. 16, 143–184, 2003.

[6] B. Parno, J. Howell, C. Gentry, and M. Raykova, Pinocchio: Nearly Practical Verifiable Computation, 2013 IEEE Symposium on Security and Privacy, 238–252, 2013.

[7] K. Sakumoto, Public-Key Identification Schemes Based on Multivariate Cubic Polynomials, In: M. Fischlin, J. Buchmann, and M. Manulis (eds), Public Key Cryptography - PKC 2012, LNCS, vol. 7293, 172–189, Springer, Berlin, Heidelberg (2012).

[8] K. Sakumoto, T. Shirai, and H. Hiwatari, Public-Key Identification Schemes Based on Multivariate Quadratic Polynomials, In: P. Rogaway (eds), Advances in Cryptology - CRYPTO 2011, LNCS, vol. 6841, 706–723, Springer, Berlin, Heidelberg (2011).

[9] V. Nachef, J. Patarin, and E. Volte, Zero-Knowledge for Multivariate Polynomials, In: A. Hevia, and G. Neven (eds), Progress in Cryptology - LATINCRYPT 2012, LNCS, vol. 7533, 194–213, Springer, Berlin, Heidelberg (2012).