令和 5 年度

修　士　論　文

# ペアリングベースアキュームレータを用いた<br>ラベル付き有向グラフにおける接続性のゼロ知識証明

先進理工系科学専攻　計算機基礎学研究室
M226526　　吉岡　卓馬

指導教員

教授　主査　中西　透
教授　副査　岩本　宙造
准教授　北須賀　輝明
教授　野上　保之（岡山大学）

令和 5 年 2 月 6 日

広島大学大学院先進理工系科学研究科

# 概要

ネットワークで接続されたシステムは, グラフとして表現することができる. システムを利用するテナントは, 自らのシステムが正しく接続されていること (接続性), 自らのシステムが他のテナントのシステムから正しく分離されていること (分離性) について, システムの提供者であるプロバイダに確認する必要がある. プロバイダは複数のテナントのシステムを管理しているため, ネットワークトポロジーを開示せずに正しい情報を証明する手法が求められている. その解決策として, ペアリングアキュームレータを用いたグラフ情報のゼロ知識証明が提案されており, 検証時間や証明データサイズがグラフの点数, 辺数に依存しないという特徴がある. しかし, この方式は 2 点の問題点がある. 1 点目に, ラベルを含めた証明が行われていないため, ネットワークの帯域やコストなどを考慮した接続性の証明ができない. 2 点目に, 無向グラフを前提としているため, ネットワークフローのような問題を扱うことができない.

本研究では, このペアリングベースの従来方式を拡張し, 各辺がラベルを持つ有向グラフにおける接続性のゼロ知識証明を提案する. さらに, 接続性に関してコストなどによる制約がある場面を想定し, テナントが作成する制約リストに含まれるラベルを持つ辺のみをパスに含む, 制約付き接続性のゼロ知識証明についても提案する. そして, これらの方式を PC 上で実装し, 処理時間およびデータサイズを測定することによって評価を行う.

# 目次

# 第 1 章

# はじめに

ネットワークで接続されたシステムは, グラフを用いて表現することができる (図 1.1). システムを利用するテナントは, 自らのシステムが正しく接続されていること (接続性), 自らのシステムが他のテナントのシステムから正しく分離されていること (分離性) について, システムの提供者であるプロバイダに確認する必要がある. しかし, プロバイダは複数のテナントのシステムを管理しているため, グラフのすべての情報を開示することができない. そのため, プロバイダはテナントに不必要な情報を一切与えずに, 接続性や分離性を満たしていることを証明する必要がある.

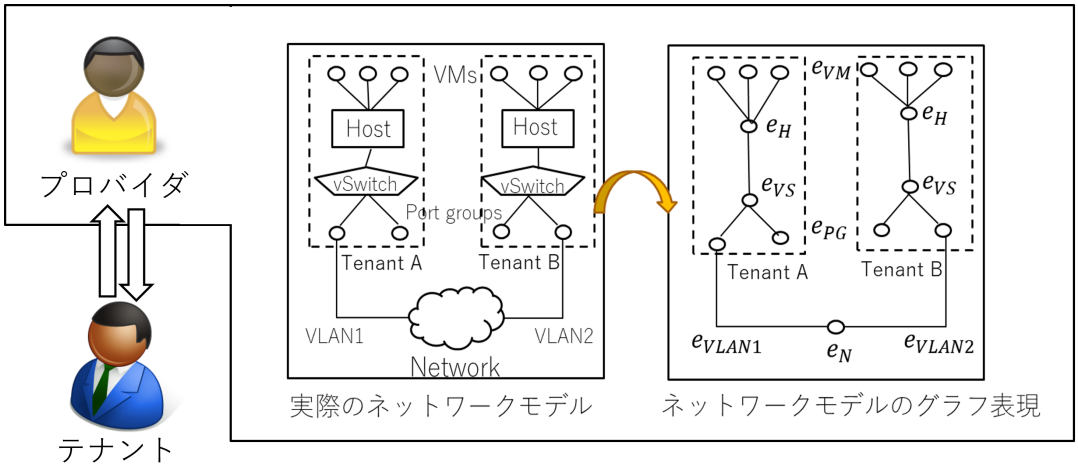


図1.1　ネットワークモデル図

この問題の解決策として, ゼロ知識証明を用いた方式が提案されている [1]. ゼロ知識証明とは, ある命題が正しいということを, それ以外の情報を伝えることなく正しいと証明

する手法である．ゼロ知識証明を利用することで, プロバイダはグラフのすべての情報を開示することなく, グラフ上の任意の 2 点における接続性, 分離性を証明することができる．このとき, グラフ情報は信頼できる認証機関による署名が付与されることにより, その正しさが保証される．

従来方式 [1] では, RSA 暗号ベースの署名とゼロ知識証明を用いて接続性, 分離性の証明を行っている．しかし, この方式は検証時間がグラフの点数や辺数に依存して大きくなるという問題点がある．この依存を改善するために, ペアリングベースのアキュームレータを用いた方式 [2] が提案されている．この方式では, グラフ上の点集合, 辺集合をアキュームレータとしてそれぞれ 1 つの値に圧縮しているため, ゼロ知識証明にかかる検証時間がグラフの点数, 辺数に依存しないという利点がある．しかし, この方式には問題点が 2 点ある．1 点目に, この方式ではグラフ中のラベルを考慮していないため, ラベルを用いて表現することができるネットワークの帯域やコストなどを考慮に入れた接続性の証明ができない．2 点目に, この方式は無向グラフを前提としているため, ネットワークフローやリンクドデータ [3] のような問題を扱うことができない．

本研究ではペアリングベースの従来方式 [2] を拡張し, 各辺がラベルを持つ有向グラフにおける接続性のゼロ知識証明を提案する．提案方式では, 辺のラベルに対しても整数を割り当てることでラベルを表現する．さらに, 辺の表現をビットシフトを用いたものに変更し, 辺の向きにより異なる値になるようにすることで, 有向辺を扱うことを可能にする．従来方式 [2] 同様, ペアリングベースのアキュームレータを利用することで, グラフの点数, 辺数, さらにラベル数への依存なしで接続性のゼロ知識証明を行うことができる．また, 接続性に関してネットワークの帯域などによる制約がある場面を想定し, テナントが作成する制約リストに含まれるラベルを持つ辺のみをパスに含むことを示す制約付き接続性のゼロ知識証明についても提案する．そして, これらの方式を PC 上で実装し, 処理時間およびデータサイズを測定することによって評価を行う．

# 第 2 章

# 数学的準備

## 2.1 双線形写像

本研究では, 双線形写像を構成可能な楕円曲線上の群を利用する. $\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T$ を素数の位数 $p$ の巡回群とし, $\mathbb{G}_1, \mathbb{G}_2$ の生成元をそれぞれ $g_1, g_2$ とする. このとき, 次に示す双線形写像 $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ を定義できる.

**双線形性**: 任意の $u \in \mathbb{G}_1, v \in \mathbb{G}_2$, および任意の $a, b \in \mathbb{Z}_p$ に対し, $e(u^a, v^b) = e(u, v)^{ab}$ が成り立つ.

**非退化性**: $e(g_1, g_2) \neq 1_{\mathbb{G}_T}$ ($1_{\mathbb{G}_T}$ は $\mathbb{G}_T$ 上の単位元)

このような双線形写像は楕円曲線上のペアリングにより実現できる.

## 2.2 安全性仮定

2.4で述べる AHO 署名は $q$-SFP 仮定に基づいており, ペアリングベースアキュームレータは $q$-SDH 仮定に基づいている.

**定義 1.** $q$-SFP($q$-Simultaneous Flexible Pairing) 仮定

全ての多項式時間アルゴリズム $\mathcal{A}$ に対して, 確率

$$\begin{aligned}&\mathsf{Pr}[(z^*, r^*, s^*, t^*, u^*, v^*, w^*)\\&\quad\leftarrow \mathcal{A}(g_z, h_z, g_r, h_r, a, \tilde{a}, b, \tilde{b}, \{(z_j, r_j, s_j, t_j, u_j, v_j, w_j)\}_{j=1}^{q})\\&\quad\wedge e(a, \tilde{a}) = e(g_z, z^*)e(g_r, r^*)e(s^*, t^*)\\&\quad\wedge e(b, \tilde{b}) = e(h_z, z^*)e(h_r, u^*)e(v^*, w^*)\\&\quad\wedge z^* \neq 1_{\mathbb{G}_2} \wedge z^* \neq \{z_j\}_{j=1}^{q}]\end{aligned}$$

は無視できる. このとき, $(g_z, h_z, g_r, h_r, a, b) \in \mathbb{G}_1, (\tilde{a}, \tilde{b}) \in \mathbb{G}_2, (s_j, v_j) \in \mathbb{G}_1^2, (z_j, r_j, t_j, u_j, w_j) \in \mathbb{G}_2^5$ であり, これらは

$$\begin{aligned}&e(a, \tilde{a}) = e(g_z, z_j)e(g_r, r_j)e(s_j, t_j)\\&\wedge e(b, \tilde{b}) = e(h_z, z_j)e(h_r, u_j)e(v_j, w_j)\end{aligned}$$

を満たす. また, $1_{\mathbb{G}_2}$ は $\mathbb{G}_2$ の単位元である.

**定義 2.** $q$-SDH($q$-Strong Diffie-Hellman) 仮定

全ての多項式時間アルゴリズム $\mathcal{A}$ に対して, $g_1, g_1^s, g_1^{s^2}..., g_1^{s^q} \in \mathbb{G}_1, g_2, g_2^s, g_2^{s^2}..., g_2^{s^q} \in \mathbb{G}_2$ を用いて, $(x, e(g_1, g_2)^{\frac{1}{s+x}})$ を出力する確率は無視できる. (ただし, $s \in \mathbb{Z}_p, x \in \mathbb{Z}_p^*\backslash\{-s\}$)

### 2.3 知識の証明

知識の証明 PoK(Proof of knowledge) とは, 証明者と検証者による対話型プロトコルであり, ある関係を満たす秘密情報を知っているということを, 秘密情報を開示することなく証明することである. 本研究では, 従来方式 [2] と同様に, Fiat-Shamir ヒューリスティックを介して PoKs を変形した知識の署名 SPKs(Signature based on Proofs of knowledge) を使用し, 以下のように離散対数の秘密情報 $x$ を知ることを示すために用いる.

$$C = g_{i_1}^{x_{j_1}} \cdots g_{i_v}^{x_{j_v}} \wedge C' = g_{i'_1}^{x_{j'_1}} \cdots g_{i'_{v'}}^{x_{j'_{v'}}}$$

このとき, $g_{i_1}^{x_1} \cdots g_{i_v}^{x_t}, C, C' \in \mathbb{G}$ であり, インデックス $i_1, ..., i_v, i'_1, ..., i'_{v'} \in \{1, ..., t\}$ はそれぞれ基底 $g_1, ..., g_t$ を指し, インデックス $j_1, ..., j_v, j'_1, ..., j'_{v'} \in \{1, ..., u\}$ は秘密情報 $x_1, ..., x_u$ を指す. この SPK は 3 つ以上の等式や, $\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T$ の異なる群の要素に対する等式に拡張が可能である.

SPK $C = g_1^{x_1} \cdots g_k^{x_k}$ は以下のように構成する．まず, $r_1, \cdots r_k \xleftarrow{\$} \mathbb{Z}_p$ を選択し, $t = g_1^{r_1} \cdots g_k^{r_k}, c = H(t, g_1, ..., g_k, C, M)$ を計算する．ここで, $H$ はハッシュ関数であり, $M$ は署名されたメッセージである．また, $s_1 = r_1 + cx_1, ..., s_k = r_k + cx_k$ とする．検証は $t' = g_1^{s_1} \cdots g_k^{s_k} C^{-c}$ を計算し, $c = H(t', g_1, ..., g_k, C, M)$ をチェックすることで行われる．

### 2.4 AHO 署名

AHO 署名 [4][5] とは複数の群要素メッセージに対して署名できる方式であり, 署名検証のペアリングの関係式をゼロ知識証明することができる．num 個のメッセージに対する AHO 署名のアルゴリズムは以下のようになる．

- **AHOKeyGen:**
  ランダムに $G_r, H_r \in \mathbb{G}_1, \mu_z, \nu_z, \{\mu_i, \nu_i\}_{1 \leq i \leq \mathsf{num}} \in \mathbb{Z}_p$ を選び, $G_z = G_r^{\mu_z}, H_z = H_r^{\nu_z}, G_i = G_r^{\mu_i}, H_i = H_r^{\nu_i}$ を計算する．次に, ランダムに $\alpha_a, \alpha_b \in \mathbb{Z}_p$ を選び, $A = e(G_r, g_2^{\alpha_a}), B = e(H_r, g_2^{\alpha_b})$ を計算する．
  AHO 署名の公開鍵を $\mathsf{pk}_{\mathsf{AHO}} = (g_1, g_2, G_r, H_r, G_z, H_z, \{G_i, H_i\}_{1 \leq i \leq \mathsf{num}}, A, B)$,
  AHO 署名の秘密鍵を $\mathsf{sk}_{\mathsf{AHO}} = (\alpha_a, \alpha_b, \mu_z, \nu_z, \{\mu_i, \nu_i\}_{1 \leq i \leq \mathsf{num}})$ としてを出力する．

- **AHOSign:**
  ランダムに選んだ $\beta, \epsilon, \eta, \iota, \kappa \in \mathbb{Z}_p$ と, $\mathsf{pk}_{\mathsf{AHO}}, \mathsf{sk}_{\mathsf{AHO}}$ を用いて, 与えられたメッセージ $(M_1, ..., M_{\mathsf{num}} \in \mathbb{G}_2^{\mathsf{num}})$ に, 以下のように署名 $\sigma = (\theta_1, ..., \theta_7)$ を生成し, 出力する．

$$\theta_1 = g_2^{\beta}, \theta_2 = g_2^{\epsilon - \mu_z \beta} \Pi_{i=1}^{\mathsf{num}} M_i^{-\mu_i}, \theta_3 = G_r^{\eta}, \theta_4 = g_2^{(\alpha_a - \epsilon)/\eta},$$
$$\theta_5 = g_2^{\iota - \nu_z \beta} \Pi_{i=1}^{\mathsf{num}} M_i^{-\nu_i}, \theta_6 = H_r^{\kappa}, \theta_7 = g_2^{(\alpha_b - \iota)/\kappa}$$

- **AHOVerify:**
  与えられたメッセージ $(M_1, ..., M_{\mathsf{num}} \in \mathbb{G}_2^{\mathsf{num}})$ と署名 $\sigma = (\theta_1, ..., \theta_7)$ が以下の検証式を満たす場合に受理する．

$$A = e(G_z, \theta_1) \cdot e(G_r, \theta_2) \cdot e(\theta_3, \theta_4) \cdot \Pi_{i=1}^{\mathsf{num}} e(G_i, M_i),$$
$$B = e(H_z, \theta_1) \cdot e(H_r, \theta_5) \cdot e(\theta_6, \theta_7) \cdot \Pi_{i=1}^{\mathsf{num}} e(H_i, M_i)$$

AHO 署名は $q$-SFP 仮定に基づき, existential unforgeability が証明されている [4], [5]. また, AHO 署名は同じメッセージに対する別の unlinkable な署名にランダマイズ可能である.

### 2.5 ペアリングベースアキュームレータ

ペアリングベースアキュームレータ [6] は, ある要素が集合に含まれていることを効率よく証明できる手法である.

- **AccSetup:**
  $\mathbb{G}_1, \mathbb{G}_2$ の生成元をそれぞれ $g_1, g_2$ として, ランダムに $s \in \mathbb{Z}_p$ を選ぶ. acc 秘密鍵 $\mathsf{sk}_{\mathsf{acc}} = s$, acc 公開鍵 $\mathsf{pk}_{\mathsf{acc}} = (g_1, g_1^{s^1}, ..., g_1^{s^q}, g_2, g_2^{s^1}, ..., g_2^{s^q})$ をそれぞれ計算し, 出力する.

- **AccGen:**
  $\mathsf{sk}_{\mathsf{acc}}, \mathsf{pk}_{\mathsf{acc}}$ を用いて, $\mathbb{Z}_p^*$ の要素の集合 $\mathcal{X}$ のアキュームレータ $\mathsf{acc}_{\mathcal{X}}$ を以下のように計算し, 出力する.
  $$\mathsf{acc}_{\mathcal{X}} = g_2^{\Pi_{x_i \in \mathcal{X}}(s+x_i)}$$
  また, $\mathsf{acc}_{\mathcal{X}} = g_2^{\Pi_{x_i \in \mathcal{X}}(s+x_i)}$ は変形すると, $\Pi_{x_i \in \mathcal{X}}(X + x_i) = \Sigma_{0 \le k \le |\mathcal{X}|} a_k X^k$ となる多項式の係数 $a_k \in \mathbb{Z}_p$ に対して, $\mathsf{acc}_{\mathcal{X}} = g_2^{\Sigma_{0 \le k \le |\mathcal{X}|} a_k s^k} = \Pi_{0 \le k \le |\mathcal{X}|} (g_2^{s^k})^{a_k}$ となるため, acc 秘密鍵 $\mathsf{sk}_{\mathsf{acc}} = s$ を用いずに計算することができる.

- **AccWitGen:**
  $\mathsf{pk}_{\mathsf{acc}}$ を用いて, 要素 $\tilde{x} \in \mathcal{X}$ に対する以下のような補助情報 $W$ を計算し, 出力する.
  $$W = g_1^{\Pi_{x_i \in \mathcal{X} \setminus \tilde{x}}(s+x_i)}$$
  この $W$ も $\mathsf{acc}_{\mathcal{X}}$ と同様に $\mathsf{sk}_{\mathsf{acc}}$ を用いずに計算できる.

- **AccVerify:**
  $\mathsf{pk}_{\mathsf{acc}}, W$ を用いて, $\tilde{x} \in \mathcal{X}$ を以下の式で検証する.
  $$e(W, g_2^s g_2^{\tilde{x}}) = e(g_1, \mathsf{acc}_{\mathcal{X}})$$

このアキュームレータは双線形 $q$-SDH 仮定に基づき安全である [6].

# 第 3 章

# 先行研究

本章では, 先行研究として [2] で提案されている, グラフに対する接続性のゼロ知識証明について紹介する. この方式はペアリングに基づいており, 群演算は楕円曲線上で計算される. 従来方式 [1] と同様にグラフの各頂点に頂点識別子として素数 $e_i$ を割り当て, 辺情報を両端の頂点識別子を掛け合わせた $e_ie_j$ と表現する. この時隣接している 2 点間の接続性は, 両端の点の値が辺の値を割り切ることを示すことで証明することができる. これを証明したい 2 点を結ぶパス上のすべての辺で行うことによって, 2 点間の接続性が証明される.

この方式では, グラフ上の点集合 $v$, 辺集合 $\varepsilon$ をそれぞれ以下のようにアキュームレータ $\mathsf{acc}_V, \mathsf{acc}_\varepsilon$ に圧縮した上で接続性の証明を行っている.

$$\mathsf{acc}_V = g_2^{\Pi_{i\in V}(s+e_i)}$$
$$\mathsf{acc}_\varepsilon = g_2^{\Pi_{(i,i')\in\varepsilon}(s+e_ie_{i'})}$$

そして証明時には, アキュームレータと署名の正しさがゼロ知識証明される. アキュームレータを用いて検証を行うことができるため, 検証時間がグラフ内の点数や辺数に依存しないことが利点である.

さらに分離性の証明を行うため, グラフ内の点集合の内, 接続されているもの同士を 1 つの連結成分と定義し, $L$ 個の連結成分に対して, それぞれの連結成分内の点集合 $V_l$ についても点集合 $V$ と同様に, アキュムレータ $\mathsf{acc}_{V_l}$ に圧縮している.

$$\mathsf{acc}_{V_l} = g_2^{\Pi_{i\in v_l}(s+e_i)}$$

そして分離性を証明する 2 点が異なるアキュームレータに存在することを示すことで, 2 点間の分離性が証明される.

この方式の問題点は 2 点ある. 1 点目は, グラフでのラベルを想定していないため, ラベルで表現される, ネットワークの帯域やコストなどの制約を含めた接続性の証明ができない点である. 2 点目は, 無向グラフを前提としており, $i, j$ 間の辺が $e_i e_j = e_j e_i$ より方向を表せず有向辺に対応できないため, ネットワークフローのような問題に適用できないという点である.

# 第 4 章

# ラベル付きグラフに対するゼロ知識証明のモデル

本論文では, 点集合 $\mathcal{V} = \{1, 2, ..., N\}$, 辺ラベル集合 $\Lambda = \{l_1, l_2, ..., l_{N'}\}$, ラベル付き辺集合 $\varepsilon = \{(i_1, i'_1, l_1), ..., (i_{\tilde{N}}, i'_{\tilde{N}}, l_{\tilde{N}})\}$ で構成されたグラフ $\mathcal{G} = (\mathcal{V}, \varepsilon, \Lambda)$ を考える. このとき, $N = |\mathcal{V}|, N' = |\Lambda|, \tilde{N} = |\varepsilon|$ である. また各 $L_j$ は $J$ 個のラベルの列と仮定する.

## 4.1 アルゴリズムの定義

グラフ署名のゼロ知識証明 GS-ZPK(Zero-knowledge proof of graph signature) は以下のアルゴリズムで構成されている.

- **IssuerKeygen:**
  セキュリティパラメータ $\lambda$ に対する $1^\lambda$, グラフにおける連結成分の最大個数 $L, N, N', \tilde{N}$ の上限 $q$ を入力として, 秘密鍵 isk, 公開鍵 ipk を生成し, 出力する.
- **Issue:**
  isk, ipk を用いて, 時刻 $t$ に対してグラフ $\mathcal{G}$ に署名 $\sigma$ を付与する.
- **ProofGen:**
  ipk, $\mathcal{G}, t, \sigma$ を用いて関係性 $R$ の証明を行う. 本論文では, $R$ は以下の二種類について考える.

  Connectivity$(t, i, j, K)$: 時刻 $t$ のグラフにおいて, 点 $i, j$ が接続されていることを示す. このとき, $K$ は $i, j$ 間のパスにおける点の数である.

  Constraint-connectivity$(t, i, j, K, \Lambda')$: 上記の Connectivity に加え, $i, j$ 間における各辺のラベルがそれぞれ, 検証者によって作成される制約リスト $\Lambda' =$

$\{l_1, l_2, ...\}$ に含まれていることを示す.

- **ProofVerify:**
  証明の結果が有効なものであれば受理, そうでなければ拒否する.

### 4.2 安全性

先行研究 [2] に基づいて, グラフのゼロ知識証明の安全性を, 以下のように定義する.

#### 4.2.1 健全性

健全性とは、証明者が証明する関係 $R$ を満たす時刻 $t$ のグラフの署名を所有していない場合, その証明者による証明は検証者によって受理されないという性質である.

ここで, GST を $(\mathcal{G}, t)$, 時刻 $t$ でグラフ $\mathcal{G}$ の署名の発行を示すペアの集合とし, 以下のような事象を考える.

$$
\begin{aligned}
&\mathsf{Exp}^{snd}_{\mathcal{A}}(\lambda, L, q):\\
&\quad (\mathsf{isk}, \mathsf{ipk}) \leftarrow \mathsf{IssuerKeygen}(1^{\lambda}, L, q)\\
&\quad \mathsf{GST} \leftarrow \emptyset\\
&\quad (t^*, R^*, \pi^*) \leftarrow \mathcal{A}^{\mathsf{Issue}}(\mathsf{ipk})\\
&\quad Return\ 1\ if\ (1) \wedge (2) \wedge (3):\\
&\qquad (1)\mathsf{ProofVerify}(\mathsf{ipk}, t^*, R^*, \pi^*) = \mathsf{accept}\\
&\qquad (2)R^*\ is\ a\ valid\ relation\\
&\qquad R^* = \mathsf{Connectivity}(t^*, i, j, K)\ or\\
&\qquad R^* = \mathsf{Constraint\text{-}Connectivity}(t^*, i, j, K, \Lambda')\\
&\qquad (3)R^*\ is\ not\ satisfied\ on\ \mathcal{G}\\
&\qquad\quad for\ any\ (\mathcal{G}, t^*) \in \mathsf{GST}\\
&\qquad\quad (i.e.,\ R^* = \mathsf{connectivity}(t^*, i, j, K)\ or\\
&\qquad\quad\ R^* = \mathsf{Constraint\text{-}connectivity}(t, i, j, K, \Lambda')\\
&\qquad\quad\ is\ not\ true\ on\ \mathcal{G}),\\
&\quad Otherwise, return\ 0
\end{aligned}
$$

この例では、次のオラクルが $\mathcal{A}$ によって呼び出される.

**Issue**： $\mathcal{A}$ はグラフ $\mathcal{G}$, 時刻 $t$ の署名を問い合わせる. このオラクルは $\sigma \leftarrow \mathsf{Issue}(\mathsf{ipk}, \mathsf{isk}, \mathcal{G}, t)$ を応答し, $(\mathcal{G}, t)$ を GST に追加する.

**定義 3** (健全性)**.**

任意のセキュリティパラメータ $\lambda \in \mathbb{N}$, 任意の $L, q \in \mathbb{N}$, および任意の PPT 敵対者 $\mathcal{A}$ に対し, $\mathsf{Pr}[\mathsf{Exp}^{snd}_{\mathcal{A}}(\lambda, L, q) = 1]$ が無視できる場合, GS-ZKP は健全である.

### 4.2.2 ゼロ知識性

ゼロ知識性とは, 検証者は公開パラメータと証明された関係以上の情報を得られないという性質である.

ここで, GST を, 時刻を $t \in \mathbb{N}$ でグラフ $\mathcal{G}$ の署名 $\sigma$ の発行を示す $(\mathcal{G}, t, \sigma)$ の集合とし, 以下のような実際のプロトコルの事象を考える.

$$
\begin{aligned}
&\mathsf{Exp}^{ZK-real}_{\mathcal{A}}(\lambda, L, q) : \\
&\quad (\mathsf{isk}, \mathsf{ipk}) \leftarrow \mathsf{IssuerKeygen}(1^{\lambda}, L, q) \\
&\quad \mathsf{GST} \leftarrow \emptyset \\
&\quad b \leftarrow \mathcal{A}^{\mathsf{Issue}, \mathsf{ProofGen}}(\mathsf{ipk})
\end{aligned}
$$

この事象では, $\mathcal{A}$ の出力は $b \in \{0, 1\}$ を満たす. また, 次のオラクルが $\mathcal{A}$ によって呼び出される.

**Issue** : $\mathcal{A}$ はグラフ $\mathcal{G}$, 時刻 $t$ の署名を問い合わせる. このオラクルは $\sigma \leftarrow \mathsf{Issue}(\mathsf{ipk}, \mathsf{isk}, \mathcal{G}, t)$ を応答し, $(\mathcal{G}, t, \sigma)$ を GST に追加する.

**ProofGen** : $\mathcal{A}$ はグラフ $\mathcal{G}$, 時刻 $t$, 関係 $R$ を問い合わせる. もし $(\mathcal{G}, t, \cdot) \notin \mathsf{GST}$ であれば停止する. $(\mathcal{G}, t, \sigma) \in \mathsf{GST}$ の場合, このオラクルは $\pi \leftarrow \mathsf{ProofGen}(\mathsf{ipk}, \mathcal{G}, t, \sigma, R)$ を返す.

次に, 以下のような理想的なプロトコルの事象を考える.

$$
\begin{aligned}
&\mathsf{Exp}^{ZK-ideal}_{\mathcal{A}}(\lambda, L, q) : \\
&\quad (\mathsf{isk}, \mathsf{ipk}) \leftarrow \mathsf{IssuerKeygen}(1^{\lambda}, L, q) \\
&\quad \mathsf{GST} \leftarrow \emptyset \\
&\quad b \leftarrow \mathcal{A}^{\mathsf{Issue}, \mathsf{ProofGenSim}}(\mathsf{ipk})
\end{aligned}
$$

この事象では, $\mathcal{A}$ の出力は $b \in \{0, 1\}$ を満たす. また, 次のオラクルが $\mathcal{A}$ によって呼び出される.

**Issue** : 上記と同様.

**ProofGenSim** : $\mathcal{A}$ はグラフ $\mathcal{G}$, 時刻 $t$, 関係 $R$ を問い合わせる. もし $(\mathcal{G}, t, cdot) \notin \mathsf{GST}$ であれば停止する. $(\mathcal{G}, t, \sigma) \in \mathsf{GST}$ の場合, このオラクルは $\pi \leftarrow \mathsf{Sim}(\mathsf{ipk}, \mathcal{G}, t, \sigma, R)$ を返す.

**定義 4** (ゼロ知識性)**.**

任意のセキュリティパラメータ $\lambda \in \mathbb{N}$, 任意の $L, q \in \mathbb{N}$, および任意の PPT 敵対者 $\mathcal{A}$ に対し, $\mathsf{Pr}[\mathsf{Exp}_{\mathcal{A}}^{ZK-real}(\lambda, L, q) = 1] - \mathsf{Pr}[\mathsf{Exp}_{\mathcal{A}}^{ZK-ideal}(\lambda, L, q) = 1]$ が無視できる場合, GS-ZKP はゼロ知識である.

# 第 5 章

# 提案方式

## 5.1 提案方式の概要

提案方式では, 各辺が $J$ 個のラベルを持つグラフにおいて, 従来方式 [2] と同様ペアリングベースでの接続性のゼロ知識証明を行う. このとき, 従来方式 [1] のように, ラベルは頂点同様にラベル識別子として整数 $e_l$ を割り当てることとする. また, 従来方式 [2] では, 辺情報を両端の点の頂点識別子の積 $e_i e_j$ と表現していたが, 提案方式ではラベル付き有向辺情報を用いるため, ビットシフトを用いて $e_i \times 2^{2b} + e_j \times 2^b + e_l$ と表現する. $b$ ビットシフトすることで, $\epsilon_{(i,j)}$ と $\epsilon_{(i,j)}$ の値が異なり, 区別することができる.

まず, 点集合 $V$, ラベル付き辺集合 $\varepsilon$, およびラベル集合 $\Lambda$ をそれぞれアキュームレータ $\mathsf{acc}_V, \mathsf{acc}_\varepsilon, \mathsf{acc}_\Lambda$ に圧縮する. そして, 生成したアキュームレータに AHO 署名を付与し, ラベルを含めたグラフ情報の正しさを保証する.

次に, 接続性を証明したい 2 点間のパス上の全ての点, 辺, および各辺のラベルについて, アキュームレータによる包含関係を示すことでグラフに含まれていることを証明する. 各辺情報が点情報, 辺情報, ラベル情報から構成できることを示すことで, 接続性を証明する.

また, 従来通りの接続性に加えて, 特定の制約下での接続性のゼロ知識証明も提案する. テナントはまず, 証明したい 2 点を結ぶパスに含めることを許容できるラベルの集合を制約リスト $\Lambda'$ として設定し, プロバイダと共有しておく. プロバイダは, 接続性を証明する 2 点間のパス上の辺ラベルが全て制約リスト $\Lambda'$ に含まれていることをアキュームレータを用いて示すことで, 制約下での接続性を証明する.

### 5.2 グラフ符号化

従来方式 [1] と同様にグラフ情報を以下のように符号化する.

1. 頂点 $i$ に対し, 頂点識別子として整数 $e_i$ を割り当てる.
2. 各ラベル $l \in \Lambda$ に対しても整数 $e_l$ を割り当てる.
3. 辺 $(i, j)$ について, ラベル付き有向辺として, 頂点識別子とラベル識別子をビットシフト, 加算した値 $e_i \times 2^{2b} + e_j \times 2^b + e_l$ で表現する. このとき, $b$ は全ての $e_i, e_l$ に対して, $e_i < 2^b$ かつ $e_l < 2^b$ かつ $e_i \times 2^{2b} + e_j \times 2^b + e_l < p$ となるように調整した値である.

### 5.3 提案プロトコル

- **IssuerKeygen:**
  グラフ情報の署名に必要な秘密鍵や公開鍵などのパラメータを生成する.
  1. $\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T$ を双線形群とし, $g_1, g_2$ を $\mathbb{G}_1, \mathbb{G}_2$ の生成元とする. また, 双線形写像 $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ として、$\mathsf{param} = (\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, g_1, g_2)$ とする.
  2. AHO 署名の秘密鍵 $\mathsf{sk}_{\mathsf{AHO}}$, 公開鍵 $\mathsf{pk}_{\mathsf{AHO}}$ をそれぞれ生成する.
  3. アキュムレータの秘密鍵 $\mathsf{sk}_{\mathsf{acc}}$, 公開鍵 $\mathsf{pk}_{\mathsf{acc}}$ をそれぞれ生成する.
  4. ランダムな $h_1 \in \mathbb{G}_1, h_2 \in \mathbb{G}_2$ を生成する.
  5. 秘密鍵 $\mathsf{isk} = (\mathsf{sk}_{\mathsf{AHO}}, \mathsf{sk}_{\mathsf{acc}})$, 公開鍵 $\mathsf{ipk} = (\mathsf{param}, \mathsf{pk}_{\mathsf{AHO}}, \mathsf{pk}_{\mathsf{acc}}, h_1, h_2)$ を出力する.
- **Issue:**
  $\mathsf{isk}, \mathsf{ipk}$ を用いて, グラフ情報をアキュムレータに圧縮し, AHO 署名を生成する.
  1. グラフ内の点集合 $V$ は, $\mathsf{sk}_{\mathsf{acc}} = s$ を用いて以下のようにアキュムレータ $\mathsf{acc}_V$ に圧縮することができる.

     $$\mathsf{acc}_V = g_2^{\prod_{i \in V}(s + e_i)}$$

     分離性の証明に関しては従来方式 [2] 同様に行えるようにするため, $L$ 個の連結成分に対して, それぞれ連結成分内の点集合 $V_l$ についても点集合 $V$ と同様にアキュムレータ $\mathsf{acc}_{V_l}$ に圧縮する.

     $$\mathsf{acc}_{V_l} = g_2^{\prod_{i \in V_l}(s + e_i)}$$

  2. グラフ内のラベル集合 $\Lambda$ についても同様にアキュームレータ $\mathsf{acc}_\Lambda$ に圧縮

する.

$$\mathsf{acc}_\Lambda = g_2^{\prod_{i\in\Lambda}(s+e_i)}$$

3. グラフ内のラベル付き辺集合 $\varepsilon$ については, すべての辺につき $J$ 個のラベル付き辺情報を 1 つのアキュームレータ $\mathsf{acc}_\varepsilon$ に圧縮する.

$$\mathsf{acc}_\varepsilon = g_2^{\prod_{(i,i',k)\in\varepsilon}(s+e_i\times 2^{2b}+e_j\times 2^b+e_k)}$$

4. $\mathsf{sk}_{\mathsf{AHO}}$ を用いて, 生成した全てのアキュームレータ $(\mathsf{acc}_V, \mathsf{acc}_{V_1}, ..., \mathsf{acc}_{V_L}, \mathsf{acc}_\Lambda, \mathsf{acc}_\varepsilon)$ をメッセージとする AHO 署名を生成する.

$$\sigma = (\theta_1, \cdots, \theta_7)$$

5. AHO 署名 $\sigma$ を出力する.

- **ProofGen:**
$\mathsf{ipk}, \sigma$ を用いて, AHO 署名と接続性の二つの知識証明を行う.

**AHO 署名の知識証明**

以下の手順で AHO 署名の知識証明を行う.

1. AHO 署名を再ランダム化し, $\sigma' = (\theta'_1, \cdots, \theta'_7)$ を得る.
2. $j = 1, 2, 5$ について, それぞれの $\theta'_j$ に対して, $\mathbb{Z}_p$ の中からランダムに選んだ $r_{\theta'_j}$ を用いてコミットメントを生成する.

$$C_{\theta'_j} = \theta'_j h^{r_{\theta'_j}}$$

3. $\mathsf{acc}_V, \mathsf{acc}_\Lambda, \mathsf{acc}_\varepsilon$ に対して, $\mathbb{Z}_p$ の中からランダムに選んだ $r_{\mathsf{acc}_V}, r_{\mathsf{acc}_\varepsilon}, r_{\mathsf{acc}_\Lambda}$ を用いて以下のようにコミットメントを計算する.

$$C_{\mathsf{acc}_V} = \mathsf{acc}_V h_2^{r_{\mathsf{acc}_V}}$$
$$C_{\mathsf{acc}_\varepsilon} = \mathsf{acc}_\varepsilon h_2^{r_{\mathsf{acc}_\varepsilon}}$$
$$C_{\mathsf{acc}_\Lambda} = \mathsf{acc}_\Lambda h_2^{r_{\mathsf{acc}_\Lambda}}$$

同様に, 連結成分内の点集合 $V_l$ に対しても同様に $\mathbb{Z}_p$ の中からランダムに選んだ $r_{\mathsf{acc}_{V_l}}$ を用いてコミットメントを生成する.

$$C_{\mathsf{acc}_{V_l}} = \mathsf{acc}_{V_l} h_2^{r_{\mathsf{acc}_{V_l}}}$$

4. AHO 署名の検証式に上記のコミットメントを代入することで, 以下の検証式が得られる.

$$
\begin{aligned}
&A^{-1} \cdot e(G_z, C_{\theta'_1}) \cdot e(G_r, C_{\theta'_2}) \cdot e(\theta'_3, \theta'_4) \cdot e(G_1, g_2^t) \cdot e(G_2, C_{\mathsf{acc}_V}) \\
&\cdot e(G_3, C_{\mathsf{acc}_\varepsilon}) \cdot e(G_4, C_{\mathsf{acc}_\Lambda}) \cdot e(G_5, C_{\mathsf{acc}_{V_l}}) \cdots e(G_{L+4}, C_{\mathsf{acc}_{V_L}}) \\
&= e(G_z, h_2)^{r_{\theta'_1}} \cdot e(G_r, h_2)^{r_{\theta'_2}} \cdot e(G_2, h_2)^{r_{\mathsf{acc}_V}} \\
&\cdot e(G_3, h_2)^{r_{\mathsf{acc}_\epsilon}} \cdot e(G_4, h_2)^{r_{\mathsf{acc}_\Lambda}} \cdot e(G_5, h_2)^{r_{\mathsf{acc}_{V_1}}} \cdots e(G_{L+4}, h_2)^{r_{\mathsf{acc}_{V_L}}} \quad (5.1)
\end{aligned}
$$

$$
\begin{aligned}
&B^{-1} \cdot e(H_z, C_{\theta'_1}) \cdot e(H_r, C_{\theta'_5}) \cdot e(\theta'_6, \theta'_7) \cdot e(H_1, g_2^t) \cdot e(H_2, C_{\mathsf{acc}_V}) \\
&\cdot e(H_3, C_{\mathsf{acc}_\varepsilon}) \cdot e(H_4, C_{\mathsf{acc}_\Lambda}) \cdot e(H_5, C_{\mathsf{acc}_{V_l}}) \cdots e(H_{L+4}, C_{\mathsf{acc}_{v_L}}) \\
&= e(H_z, h_2)^{r_{\theta'_1}} \cdot e(H_r, h_2)^{r_{\theta'_5}} \cdot e(H_2, h_2)^{r_{\mathsf{acc}_V}} \\
&\cdot e(H_3, h_2)^{r_{\mathsf{acc}_\epsilon}} \cdot e(H_4, h_2)^{r_{\mathsf{acc}_\Lambda}} \cdot e(H_5, h_2)^{r_{\mathsf{acc}_{V_1}}} \cdots e(H_{L+4}, h_2)^{r_{\mathsf{acc}_{V_L}}} \quad (5.2)
\end{aligned}
$$

この 2 つの式に対して知識の証明を行う.

**接続性の知識証明**

$(i_1, \cdots, i_K)$ を頂点 $i(i_1 = i)$ から頂点 $j(i_K = j)$ までのパスとし, $1 \leq k \leq K$ としたときのそれぞれの辺 $(i_k, i_{k+1})$ に対して、二つの頂点の間にパスが存在することを証明する.

1. $2 \leq k \leq K-1$ に対して, コミットされた $e_{i_k}$ が $\mathsf{acc}_v$ に含まれていることを示す.

   まず, 補助情報 $W_{i_k}$ を以下のように計算する.

$$W_{i_k} = g_1^{\Pi_{t \in V, t \neq i_k}(s+e_t)}$$

   補助情報 $W_{i_k}$ に対して, $\mathbb{Z}_p$ の中からランダムに選んだ $r_{W_{i_k}}$ を用いてコミットメントを生成する.

$$C_{W_{i_k}} = W_{i_k} h_1^{r_{W_{i_k}}}$$

   頂点 $e_{i_k}$ に対して, $\mathbb{Z}_p$ の中からランダムに選んだ $r_{e_{i_k}}$ を用いてコミットメントを生成する.

$$C_{e_{i_k}} = g_1^{e_{i_k}} h_1^{r_{e_{i_k}}}$$

   ここで,

$$\alpha_k = e_{i_k} r_{W_{i_k}}$$

として, $\mathbb{Z}_p$ の中からランダムに選んだ $r_{\alpha_k}$ を用いてコミットメントを生成する.

$$C_{\alpha_k} = g_1^{\alpha_k} h_1^{r_{\alpha_k}}$$

さらに, $\beta_k$ を以下のように計算する.

$$\beta_k = r_{\alpha_k} - r_{e_{i_k}} r_{W_{i_k}}$$

上記を用いると, 以下の検証式が得られる.

$$\begin{aligned} & e(C_{W_{i_k}}, g_2^s) \cdot e(g_1, C_{\mathsf{acc}_V})^{-1} \\ & = e(C_{W_{i_k}}, g_2^{-1})^{e_{i_k}} \cdot e(h_1, g_2)^{\alpha_k} \cdot e(h_1, g_2^s)^{r_{W_{i_k}}} \cdot e(g_1, h_2^{-1})^{r_{\mathsf{acc}_V}} \end{aligned} \tag{5.3}$$

$$C_{e_{i_k}} = g_1^{e_{i_k}} h_1^{r_{e_{i_k}}}, C_{\alpha_k} = g_1^{\alpha_k} h_1^{r_{\alpha_k}} \tag{5.4}$$

$$C_{\alpha_k} = C_{e_{i_k}}^{r_{W_{i_k}}} h_1^{\beta_k} \tag{5.5}$$

これらの式に対して知識の証明を行う.

2. 各辺の $e_l$ ついて, 以下のいずれかを行う.
   (a) 制約リストを用いない場合 ($R = \mathsf{Connectivity}(t, i, j, K)$)
   $1 \leq k \leq K-1$ に対して, $k$ 番目の辺のラベルがコミットされた値 $e_{l_k}$ が $\mathsf{acc}_\Lambda$ に含まれていることを示す.
   まず, 補助情報 $W_{l_k}$ を以下のように計算する.

$$W_{l_k} = g_1^{\prod_{t \in \Lambda, t \neq l_k}(s + e_t)}$$

   補助情報 $W_{l_k}$ に対して, $\mathbb{Z}_p$ の中からランダムに選んだ $r_{W_{l_k}}$ を用いてコミットメントを生成する.

$$C_{W_{l_k}} = W_{i_k} h_1^{r_{W_{l_k}}}$$

   ラベル $e_{l_K}$ に対して, $\mathbb{Z}_p$ の中からランダムに選んだ $r_{e_{l_k}}$ を用いてコミットメントを生成する.

$$C_{e_{l_k}} = g_1^{e_{l_k}} h_1^{r_{e_{l_k}}}$$

   ここで,

$$\alpha_k = e_{l_k} r_{W_{l_k}}$$

として, $\mathbb{Z}_p$ の中からランダムに選んだ $r_{\alpha_k}$ を用いてコミットメントを生成する.

$$C_{\alpha_k} = g_1^{\alpha_k} h_1^{r_{\alpha_k}}$$

さらに, $\beta_k$ を以下のように計算する.

$$\beta_k = r_{\alpha_k} - r_{e_{l_k}} r_{W_{l_k}}$$

上記を用いると, 以下の検証式が得られる.

$$\begin{aligned} & e(C_{W_{l_k}}, g_2^s) \cdot e(g_1, C_{\mathsf{acc}_\Lambda})^{-1} \\ & = e(C_{W_{l_k}}, g_2^{-1})^{e_{l_k}} \cdot e(h_1, g_2)^{\alpha_k} \cdot e(h_1, g_2^s)^{r_{W_{l_k}}} \cdot e(g_1, h_2^{-1})^{r_{\mathsf{acc}_\Lambda}} \end{aligned} \tag{5.6}$$

$$C_{e_{l_k}} = g_1^{e_{l_k}} h_1^{r_{e_{l_k}}}, C_{\alpha_k} = g_1^{\alpha_k} h_1^{r_{\alpha_k}} \tag{5.7}$$

$$C_{\alpha_k} = C_{e_{l_k}}^{r_{W_{l_k}}} h_1^{\beta_k} \tag{5.8}$$

これらの検証式に対して知識の証明を行う.

(b) 制約リストを用いる場合 ($R = \mathsf{Constraint\text{-}connectivity}(t, i, j, K, \Lambda')$)
検証者は事前に, 点 $i, j$ を結ぶパスに含めることを許容できるラベルの集合である, 制約リスト $\Lambda' = \{e_{l_1}, e_{l_2}, ...\}$ を作成し, 証明者と共有しているものとする.
また, 証明者, 検証者はともに, 制約リスト $\Lambda'$ をアキュームレータ $\mathsf{acc}_{\Lambda'}$ に圧縮しておく.

$$\mathsf{acc}_{\Lambda'} = g_2^{\prod_{i \in \Lambda'}(s + e_i)}$$

このとき, $1 \leq k \leq K-1$ に対して, $k$ 番目の辺のラベルがコミットされた値 $e_{l_k}$ が $\mathsf{acc}_{\Lambda'}$ に格納されていることを示す.
補助情報 $W_{l_k}$ を以下のように計算する.

$$W_{l_k} = g_1^{\prod_{t \in \Lambda', t \neq l_k}(s + e_t)}$$

上記を用いると, 以下の検証式が得られる.

$$\begin{aligned}
&e(W_{l_k}, g_2^s) \cdot e(g_1, \mathsf{acc}_{\Lambda'})^{-1} \\
&= e(g_1^{\Pi_{t\in\Lambda', t\neq l}(s+e_t)}, g_2^s) \cdot e(g_1, g_2^{\Pi_{t\in\Lambda'}(s+e_t)})^{-1} \\
&= e(g_1, g_2^{s+e_{l_k}})^{\Pi_{t\in\Lambda', t\neq l}(s+e_t)} \\
&\quad \cdot e(g_1, g_2^{-e_{l_k}})^{\Pi_{t\in\Lambda', t\neq l}(s+e_t)} \\
&\quad \cdot e(g_1, g_2)^{-\Pi_{t\in\Lambda'}(s+e_t)} \\
&= e(W_{l_k}, g_2^{-1})^{e_{l_k}} \qquad (5.9)
\end{aligned}$$

これらの検証式に対して知識の証明を行う.

3. $1 \leq k \leq K-1$ に対して, コミットされた $\epsilon_k = e_{i_k} \times 2^{2b} + e_{i_{k+1}} \times 2^b + e_{l_k}$ が $\mathsf{acc}_\varepsilon$ に含まれていることを示す。
まず, 補助情報 $W_{\epsilon_k}$ を以下のように計算する.

$$W_{\epsilon_k} = g_1^{\Pi_{(t,t')\in\varepsilon, (t,t')\neq(i_k,i_{k+1})}(s+e_t\times 2^{2b}+e_{t'}\times 2^b+e_l)}$$

補助情報 $W_{\epsilon_k}$ に対して, $\mathbb{Z}_p$ の中からランダムに選んだ $r_{W_{\epsilon_k}}$ を用いてコミットメントを生成する.

$$C_{W_{\epsilon_k}} = W_{\epsilon_k} h_1^{r_{W_{\epsilon_k}}}$$

辺 $\epsilon_k$ に対して, $\mathbb{Z}_p$ の中からランダムに選んだ $r_{\epsilon_k}$ を用いてコミットメントを生成する.

$$C_{\epsilon_k} = g_1^{\epsilon_k} h_1^{r_{W_{\epsilon_k}}}$$

ここで,

$$\alpha_k = \epsilon_k r_{W_{\epsilon_k}}$$

として, $\mathbb{Z}_p$ の中からランダムに選んだ $r_{\alpha_k}$ を用いてコミットメントを生成する.

$$C_{\alpha_k} = g_1^{\alpha_k} h_1^{r_{\alpha_k}}$$

さらに, $\beta_k$ を以下のように計算する.

$$\beta_k = r_{\alpha_k} - r_{\epsilon_k} r_{W_{\epsilon_k}}$$

上記を用いると, 以下の検証式が得られる.

$$\begin{aligned}
&e(C_{W_{\epsilon_k}}, g_2^s) \cdot e(g_1, C_{\mathsf{acc}_\varepsilon})^{-1} \\
&= e(C_{W_{\epsilon_k}}, g_2^{-1})^{\epsilon_k} \cdot e(h_1, g_2)^{\alpha_k} \cdot e(h_1, g_2^s)^{r_{W_{\epsilon_k}}} \cdot e(g_1, h_2^{-1})^{r_{\mathsf{acc}_\epsilon}} \qquad (5.10)
\end{aligned}$$

$$C_{\epsilon_k} = g_1^{\epsilon_k} h_1^{r_{\epsilon_k}}, C_{\alpha_k} = g_1^{\alpha_k} h_1^{r_{\alpha_k}} \tag{5.11}$$

$$C_{\alpha_k} = C_{\epsilon_k}^{r_{W_{\epsilon_k}}} h_1^{\beta_k} \tag{5.12}$$

これらの検証式に対して知識の証明を行う.

4. $2 \leq k \leq K-2$ に対して, $\epsilon_k = e_{i_k} \times 2^{2b} + e_{i_{k+1}} \times 2^b + e_l$ が接続していることを示す.

   1, 2, 3から, 以下のコミットメントが得られている.

$$\begin{aligned} C_{e_{i_k}} &= g_1^{e_{i_k}} h_1^{r_{e_{i_k}}} \\ C_{e_{i_{k+1}}} &= g_1^{e_{i_{k+1}}} h_1^{r_{e_{i_{k+1}}}} \\ C_{e_{l_k}} &= g_1^{e_l} h_1^{r_{e_{l_k}}} \\ C_{\epsilon_k} &= g_1^{\epsilon_k} h_1^{r_{\epsilon_k}} \end{aligned}$$

   $\gamma_k$ を以下のように計算する.

$$\gamma_k = r_{\epsilon_k} - 2^{2b} r_{e_{i_k}} - 2^b r_{e_{i_{k+1}}} - r_{e_{l_k}}$$

   上記を用いると, 以下の検証式が得られる.

$$C_{\epsilon_k} = C_{e_{i_k}}^{2^{2b}} C_{e_{i_{k+1}}}^{2^b} C_{e_{l_k}} h_1^{\gamma_k} \tag{5.13}$$

   この検証式に対して知識の証明を行う.

5. パスの両端, すなわち $\epsilon_1 = e_{i_1} \times 2^{2b} + e_{i_2} \times 2^b + e_{l_k}, \epsilon_{K-1} = e_{K-1} \times 2^{2b} + e_{i_K} \times 2^b + e_{l_k}$ が接続していることを示す.

   1, 2, 3から, 以下のコミットメントが得られている.

$$\begin{aligned} C_{\epsilon_1} &= g_1^{\epsilon_1} h_1^{r_{\epsilon_1}} \\ C_{\epsilon_{K-1}} &= g_1^{\epsilon_{K-1}} h_1^{r_{\epsilon_{K-1}}} \end{aligned}$$

   $\gamma_k$ を以下のように計算する.

$$\begin{aligned} \gamma_1 &= r_{\epsilon_1} - 2^b r_{e_{i_2}} - r_{e_{l_1}} \\ \gamma_{K-1} &= r_{\epsilon_{K-1}} - 2^{2b} r_{e_{K-1}} - r_{e_{l_{K-1}}} \end{aligned}$$

   上記を用いると, 以下の検証式が得られる.

$$C_{\epsilon_1} = g_1^{e_{i_1} 2^{2b}} C_{e_{i_2}}^{2^b} C_{e_{l_k}} h_1^{\gamma_1} \tag{5.14}$$

$$C_{\epsilon_{K-1}} = C_{e_{K-1}}^{2^{2b}} g_1^{{e_i}_K 2^b} C_{e_{l_k}} h_1^{\gamma_{K-1}} \tag{5.15}$$

これらの検証式に対して知識の証明を行う.

6. 証明 $\pi = (\{\theta'_n\}_{n=3,4,6,7}, \{C_{\theta'_l}\}_{n=1,2,5}, C_{\mathsf{acc}_V}, C_{\mathsf{acc}_\Lambda}, C_{\mathsf{acc}_\varepsilon}, \{C_{\mathsf{acc}_{V_l}}\}_{1\leq l\leq L}, V_{\mathsf{AHO}}, J,$
$\{C_{W_{i_k}}, C_{e_{i_k}}, C_{\alpha_k}, V_{e_{i_k}\in V}\}_{2\leq k\leq K-1}, \{C_{W_{l_k}}, C_{e_{l_k}}, C_{\alpha''_k}, V_{e_{l_k}\in\Lambda}\}_{1\leq k\leq K-1},$
$\{C_{W_{\epsilon_k}}, C_{\epsilon_k}, C_{\alpha'_k}, V_{\epsilon_k\in\varepsilon}\}_{1\leq k\leq K-1}, \{V_{\epsilon_k}\}_{2\leq k\leq K-2}, V_{\epsilon_1,\epsilon_{K-1}})$ を出力する.

- **ProofVerify:**
  全ての SPK を検証し, 有効でないものがあれば拒否, それでなければ受理する.

# 第 6 章

# 安全性

本章では, 以下の補題を利用して健全性, ゼロ知識性を証明する. また, 補題 1 から 4 の証明に関しては [2] で示されているため, 割愛する.

**補題 1.** $SPK\ V_{\mathsf{AHO}}$ は再ランダム化された AHO 署名 $(\theta'_1, ..., \theta'_7)$ とそのメッセージ $m_1 = g_2^t, m_2 = C_{\mathsf{acc}_\mathcal{V}} h_2^{-r_{\mathsf{acc}_\mathcal{V}}}, m_3 = C_{\mathsf{acc}_\varepsilon} h_2^{-r_{\mathsf{acc}_\varepsilon}}, m_4 = C_{\mathsf{acc}_\Lambda} h_2^{-r_{\mathsf{acc}_\Lambda}}, m_5 = C_{\mathsf{acc}_{\mathcal{V}_1}} h_2^{-r_{\mathsf{acc}_{\mathcal{V}_1}}}, ..., m_{L+4} = C_{\mathsf{acc}_{\mathcal{V}_L}} h_2^{-r_{\mathsf{acc}_{\mathcal{V}_L}}}$ の知識の証明である.

**補題 2.** $SPK\ V_{e_{i_k} \in \mathcal{V}}$ は $e(W_{i_k}, g_2^{(s+e_{i_k})}) = e(g_1, \mathsf{acc}_\mathcal{V})$ を満たす $e_{i_k}, W_{i_k}, \mathsf{acc}_\mathcal{V}$ の知識の証明である.

**補題 3.** $SPK\ V_{\epsilon_k \in \varepsilon}$ は $e(W_{\epsilon_k}, g_2^{(s+\epsilon_k)}) = e(g_1, \mathsf{acc}_\varepsilon)$ を満たす $\epsilon_k, W_{\epsilon_k}, \mathsf{acc}_\varepsilon$ の知識の証明である.

**補題 4.** $SPK\ V_{e_{l_k} \in \Lambda}$ は $e(W_{e_{l_k}}, g_2^{(s+e_{l_k})}) = e(g_1, \mathsf{acc}\ )$ を満たす $e_{l_k}, W_{e_{l_k}}, \mathsf{acc}_\Lambda$ の知識の証明である. また, $SPK\ V_{e_{l_k} \in \Lambda'}$ は $e(W_{e_{l_k}}, g_2^{(s+e_{l_k})}) = e(g_1, \mathsf{acc}\ ')$ を満たす $e_{l_k}, W_{e_{l_k}}, \mathsf{acc}'_\Lambda$ の知識の証明である.

**補題 5.** $SPK\ V_{\epsilon_k}$ は $2 \leq k \leq K-2$ において $\epsilon_k = e_{i_k} \times 2^{2b} + e_{i_{k+1}} \times 2^b + e_{l_k} (\mathsf{mod}\ p)$ を満たす $\epsilon_k, e_{i_k}, e_{i_{k+1}}, e_{l_k}$ の知識の署名である. また, $SPK\ V_{\epsilon_1, \epsilon_{K-1}}$ は $\epsilon_1 = e_{i_1} \times 2^{2b} + e_{i_2} \times 2^b + e_{l_1} (\mathsf{mod}\ p)$ を満たす $\epsilon_1, e_{i_1}, e_{i_k}, e_{l_1}$ の, $\epsilon_{K-1} = e_{i_{K-1}} \times 2^{2b} + e_{i_K} \times 2^b + e_{l_{K-1}} (\mathsf{mod}\ p)$ を満たす $\epsilon_{K-1}, e_{i_{K-1}}, e_{i_K}, e_{l_{K-1}}$ の知識の署名である.

(証明)
SPK より, $C_{\epsilon_k} = g_1^{\epsilon_k} h_1^{r_{\epsilon_k}}$ なる $\epsilon_k, r_{e_k}$ を抽出できる.

一方,

$$\begin{aligned}C_{\epsilon_k} &= C_{e_{i_k}}^{2^{2b}} C_{e_{i_{k+1}}}^{2^b} C_{e_{l_k}} h_1^{\gamma_k} \\ &= (g_1^{e_{i_k}} h_1^{r_{e_{i_k}}})^{2^{2b}} (g_1^{e_{i_{k+1}}} h_1^{r_{e_{i_{k+1}}}})^{2^b} g_1^{e_{l_k}} h_1^{r_{e_{l_k}}} h_1^{r_{\epsilon_k} - 2^{2b} r_{e_{i_k}} - 2^b r_{e_{i_{k+1}}} - r_{e_{l_k}}} \\ &= g_1^{e_{i_k} \times 2^{2b} + e_{i_{k+1}} \times 2^b + e_{l_k}} h_1^{r_{\epsilon_k}}\end{aligned}$$

を満たす $e_{i_k}, e_{i_{k+1}}, e_{l_k}$ を抽出できる. よって, 離散対数仮定から, $\epsilon_k = e_{i_k} \times 2^{2b} + e_{i_{k+1}} \times 2^b + e_{l_k} (\mathsf{mod}\ p)$ となる. $k = 1, k = K - 1$ の場合も同様である.

**定理 1.** AHO 署名とアキュームレータが安全であるならば, 提案手法は健全性を満たす.

従来方式 [2] において示されている証明の概略を以下に示す.

敵対者 $\mathcal{A}$ に対し, $\mathsf{Pr}[\mathsf{Exp}_{\mathcal{A}}^{snd}(\lambda, L, q) = 1]$ が無視できないと仮定すると, $\mathcal{A}$ は偽造した証明 $\pi^*$ を出力する. 補題 1 より, $\pi^*$ の SPK から再ランダム化した AHO 署名 $(\theta_1'^*, ..., ...)$ を抽出できる. また, パス上の $e_{i_1}^*, ..., e_{i_K}^*$ と $\epsilon_1^*, ..., \epsilon_K^*$ も抽出できる. このとき, 2 つの場合が考えられる.

まず, $\mathcal{A}$ が AHO 署名 $(\theta_1'^*, ..., \theta_7'^*)$ を偽造する場合を考える. この場合, $\mathcal{A}$ を用いて AHO 署名方式に対する敵対者 $\mathcal{A}_{\mathsf{AHO}}$ を構成できる。そうでなければ, 抽出された AHO 署名は正しく発行されるので, 署名されたアキュームレータは正しい. この場合, 以下のように $\mathcal{A}$ を用いて, アキュームレータに対する敵対者 $\mathcal{A}_{\mathsf{acc}}$ を構成できる. 上記の補題より, 各 $e_{i_k}^*$ と $\epsilon_k^*$ のアキュムレータ検証が成り立ち, $\epsilon_k^* = e_{i_k}^* \cdot e_{i_{k+1}}^* \ (\mathsf{mod}\ q)$ が成り立つ. さらに, すべての $k$ で $e_{i_k}^* \in V$ かつ $\epsilon_k^* \in \varepsilon$ と仮定すると, $\epsilon_k^* = e_{i_k}^* \cdot e_{i_{k+1}}^*$ となる. この式は健全性ゲームの勝利条件, すなわち, 対象の頂点がつながっていないことに矛盾する. したがって, $e_k^* \notin V$ または $\epsilon_k^* \notin \varepsilon$ となり, これはアキュムレータの攻撃成功を意味する.

提案方式では, ラベル付き有向辺に適応するように辺の符号化を修正している. ラベル識別子 $e_l$ は頂点と辺と同様にアキュムレータに圧縮し, アキュムレータの検証式が証明されている. 辺の向きについては, 各辺は $\epsilon_k = e_{i_k} \times 2^{2b} + e_{i_{k+1}} \times 2^b + e_l$ として符号化されれており, 辺 $(e_i, e_{i'})$ の符号化と辺 $(e_{i'}, e_i)$ の符号化は異なる. SPK において, $\epsilon_k^* = e_{i_k}^* \times 2^{2b} + e_{i_{k+1}}^* \times 2^b + e_l^* \ (\mathsf{mod}\ p)$ となることを証明されるが, 前論文と同様の考察により, この式は整数式として成り立つ. 従って, 従来方式 [2] と同様に, $\mathcal{A}_{\mathsf{acc}}$ を構成することができる. また, $R^* =$**Constraint-connectivity**$(t^*, i^*, j^*, K^*, \Lambda'^*)$ の場合, $\Lambda$ を $\Lambda'$ に置き換えることで, 上記の証明を適用できる.

**定理 2.** ランダムオラクルモデルにおいて, 提案手法はゼロ知識性を満たす.

元の論文と同様に, AHO 署名の再ランダム化, コミットメントによる完全な秘匿化, および SPK がゼロ知識性を持つことから, **ProofGen** のシミュレータを構築することができる.

# 第 7 章

# 実装・実験結果

本研究では, 表 7.1に示す実装環境で提案方式の実装, および処理時間の測定を行った.

## 7.1 実装環境

証明時間, 検証時間, アキュームレータの補助情報の計算時間, 全体の処理時間の 4 項目について計測を行った. グラフは 3 分木状 (図 7.2) であり, 特に断りがない限り, 点は 100 個, 辺は 99 個, ラベルは各辺に 1 つずつ含まれており, 証明する 2 点間の辺数は 4 として計測している. また, ラベルの値は重複があり, グラフ全体のラベル数 $|\Lambda|$ は $\frac{|\varepsilon|}{3}$ としている.

表7.1 実装環境

| | |
|---|---|
| OS | WSL2(Ubuntu 20.04.4 LTS) |
| CPU | Intel(R)Core(TM)i7-11700(2.50GHz) |
| メモリ | 16.0GB |
| プログラミング言語 | C 言語 |
| 多倍長ライブラリ | GMP6.2.0 |
| ペアリングライブラリ | ELiPS[7] |

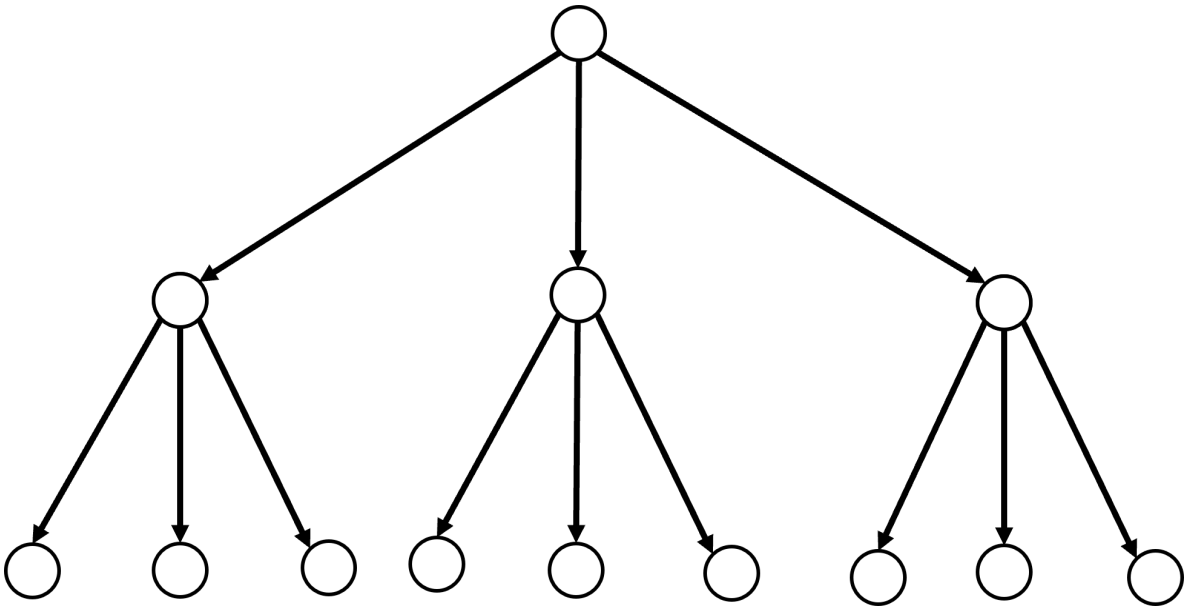

図7.2　グラフモデル図

### 7.2　点数の変化による処理時間の変化

グラフ全体の点数を 50 個から 500 個まで 50 個ずつ増加させたときの処理時間の変化を検証した (図 7.3).

このとき, 点と同時に辺とラベルの数も同様に増加している. そしてそれらの増加に伴い, 証明時間が線形で増加していることがわかる. 点情報, 辺情報, ラベル情報の補助情報 $W_{i_k}, W_{\epsilon_k}, W_{e_{l_k}}$ の計算時間がグラフ全体の点数, 辺数, ラベル数にそれぞれ比例しているためである. しかし, 検証時間は点数に関わらず一定である. これは, 従来方式 [2] と同様, 検証式にはグラフの点集合, 辺集合, ラベル集合をそれぞれ 1 つの値にまとめたアキュームレータを用いており, 検証式の数が点数, 辺数, ラベル数に依存しないためである.

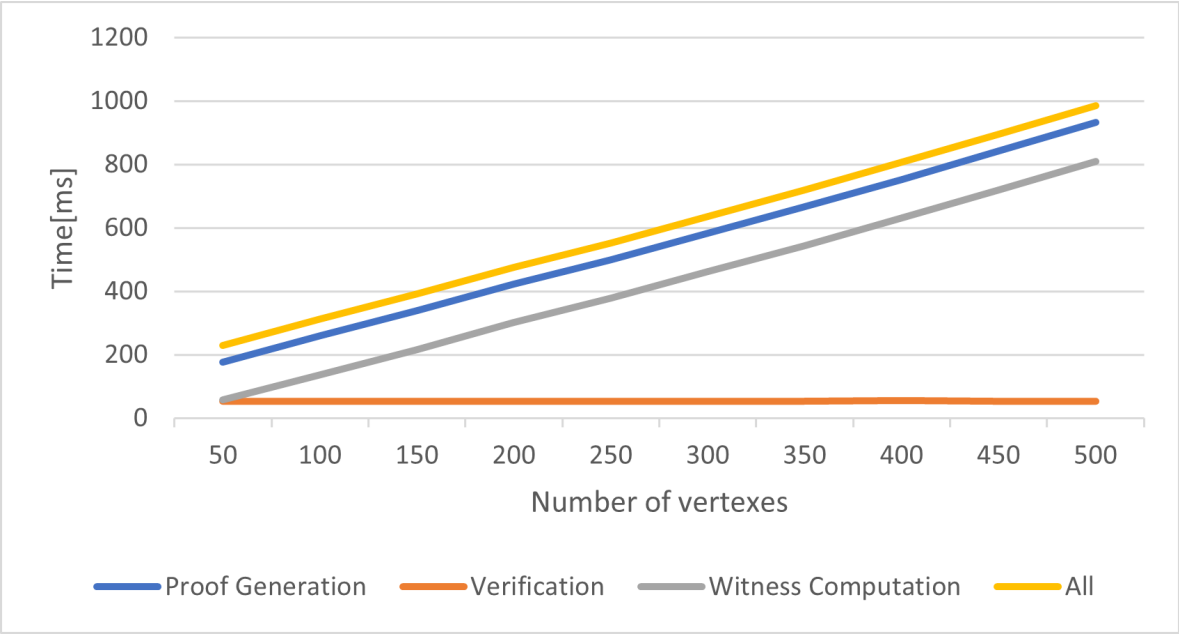


図7.3　点数の変化による処理時間の変化

### 7.3　ラベル数の変化による処理時間の変化

各辺のラベル数を 1 個から 5 個まで, すなわちグラフ全体のラベル数を 33 個から 165 個まで増加させたときの処理時間の変化を検証した (図 7.4).

この計測ではラベルの増加に伴い, ラベルのアキュームレータの要素に加えて辺のアキュームレータの要素も増加する. この方式ではラベル情報を含めた辺情報を辺情報アキュームレータの要素としているので, 各辺のラベルが増加すると, 別の要素としてアキュームレータに含められるためである. すなわち, ラベルを増加することで辺とラベルの補助情報 $W_{\epsilon_k}, W_{el_k}$ の計算時間がそれぞれ増加する. 実際に図 7.4を見てみると. 図 7.3と同様にラベル数の増加に応じて証明時間が増加しているが, 検証時間はラベル数によらず一定である.

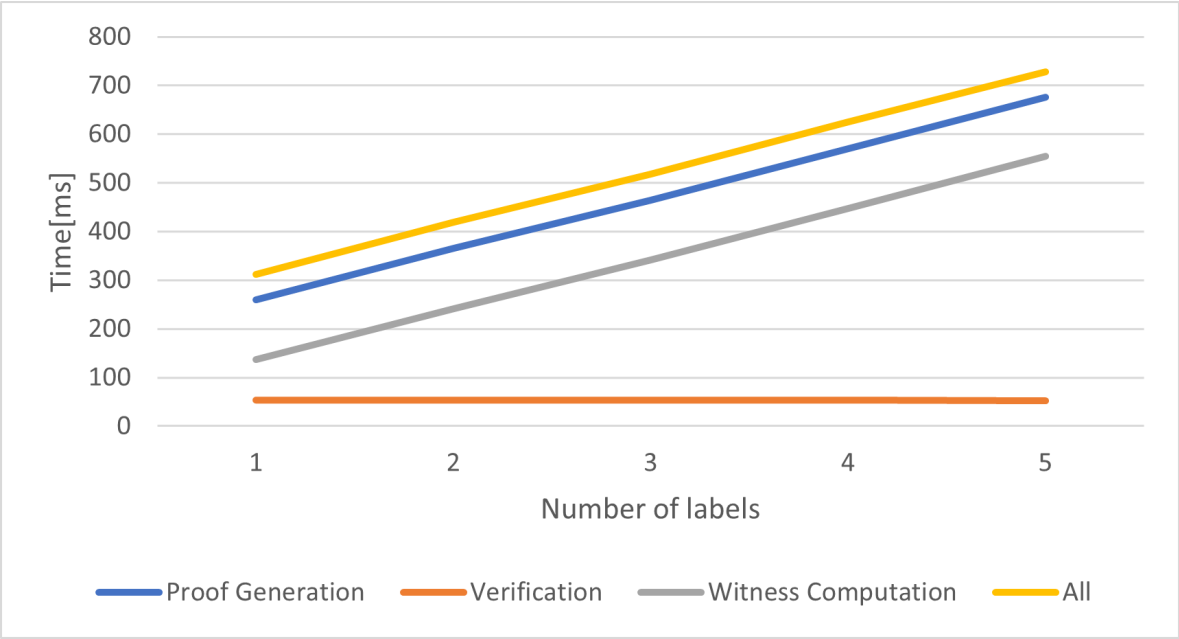


図7.4　ラベル数の変化による処理時間の変化

## 7.4　制約リストのサイズの変化による処理時間の変化

制約を含む接続性の証明について, 制約リスト内のラベル数を 10 個から 50 個まで変化させたときの処理時間の変化を検証した (図 7.5).

制約リスト内のラベル数による処理時間の変化はほとんどないことがわかる. これは点数や辺数が約 100 個あるのに対し, 制約ラベル数が 10~50 個と少ないことから, 他の処理時間に対してラベルの処理時間の割合が小さいためである.

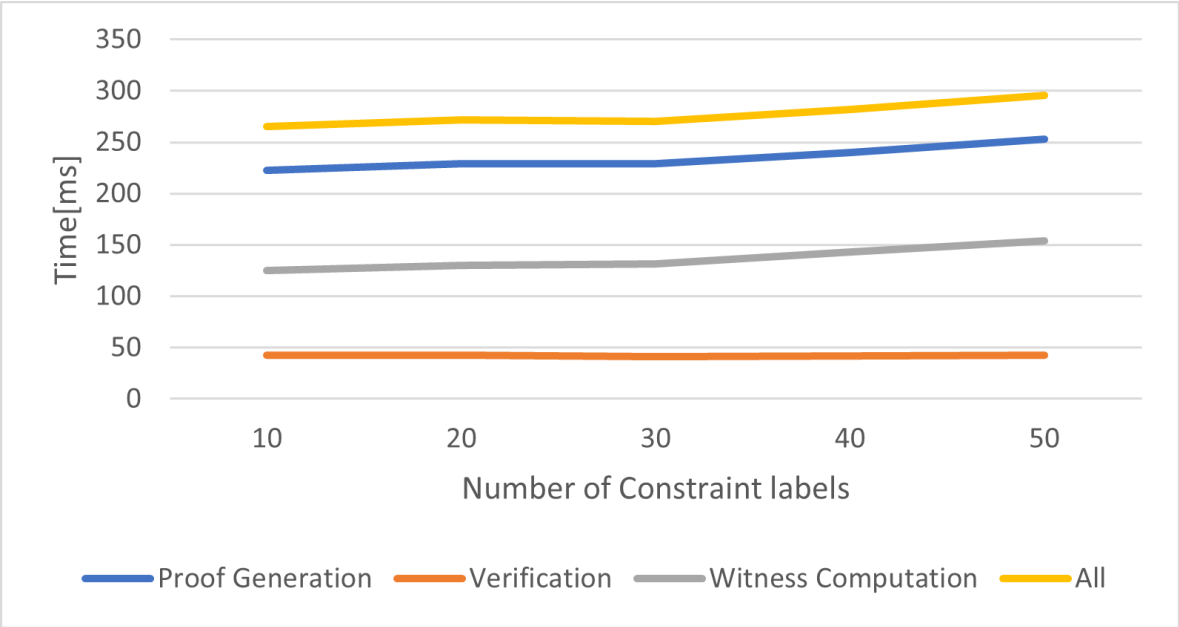


図7.5　制約リストのサイズの変化による処理時間の変化

### 7.5　証明する 2 点間の距離の変化による処理時間の変化

証明する 2 点間の点数を 1 個から 4 個まで増加させたときの処理時間の変化を検証した (図 7.6).

いずれの項目においても距離の増加に伴い処理時間が増加している．これは, **ProofGen** の内, 接続性の知識証明の 1から 4までの処理と **ProofVerify** での対応した検証処理を, 証明する 2 点間の全ての辺に対して行う必要があるためである.

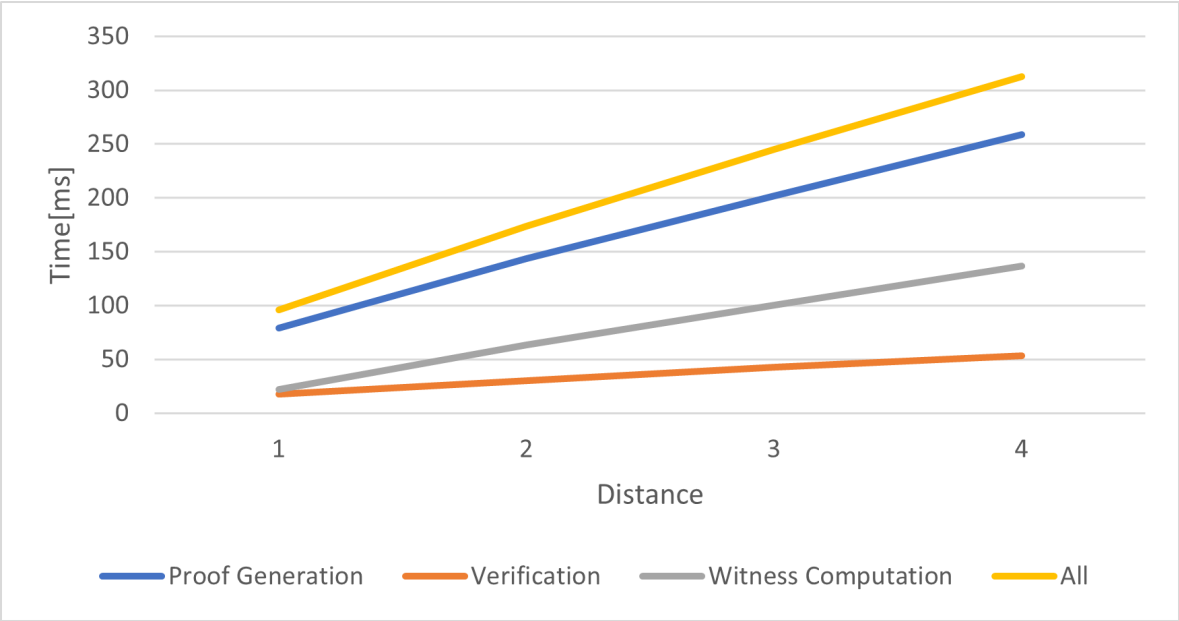


図7.6　証明する 2 点間の距離の変化による処理時間の変化

## 7.6　従来方式との比較

本研究で提案する方式と, ラベルを含まない方式である従来方式 [2] を比較することで, ラベルに関する証明, 検証を追加したことによる処理時間の変化を検証した (図 7.7).

全ての項目において処理時間が 10~20% 程度増加していることが図から読み取れる. これは, ラベルを追加したことにより, 証明, 検証ともに計算量が増えたためである. しかし, ラベルの総数は点や辺に比べて少ないため, 増加量はわずかである.

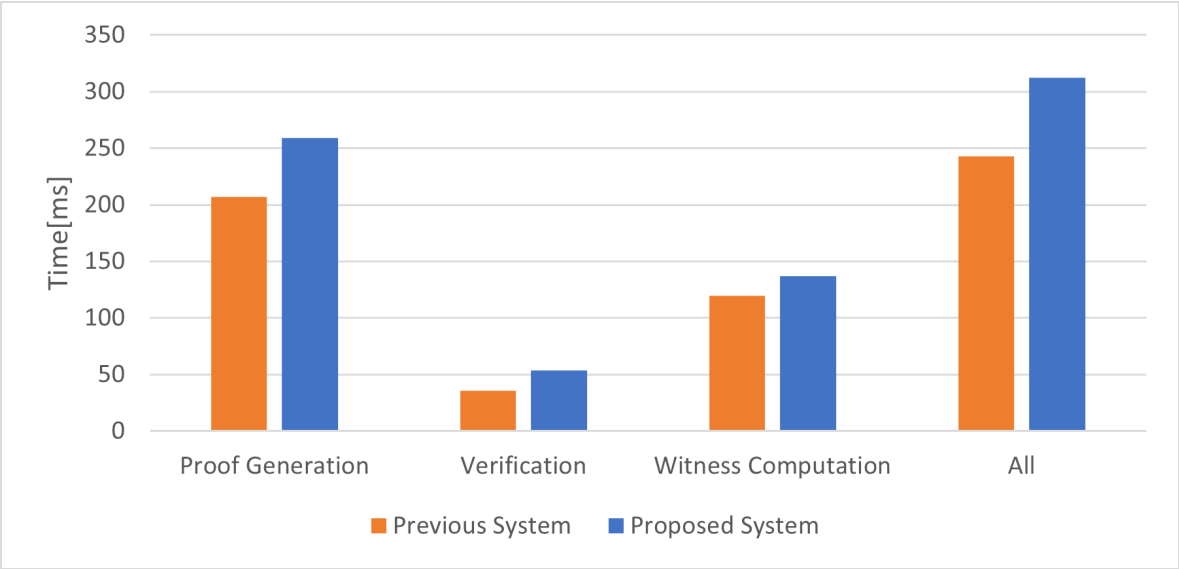


図7.7　従来方式との比較

## 7.7　伝送データサイズ

証明する 2 点間の点数を 1 個から 4 個まで増加させたときの伝送データサイズの変化 (図 7.8), 及び点数 (図 7.9), ラベル数 (図 7.10) を増加させたときの伝送データサイズの変化をそれぞれを検証した.

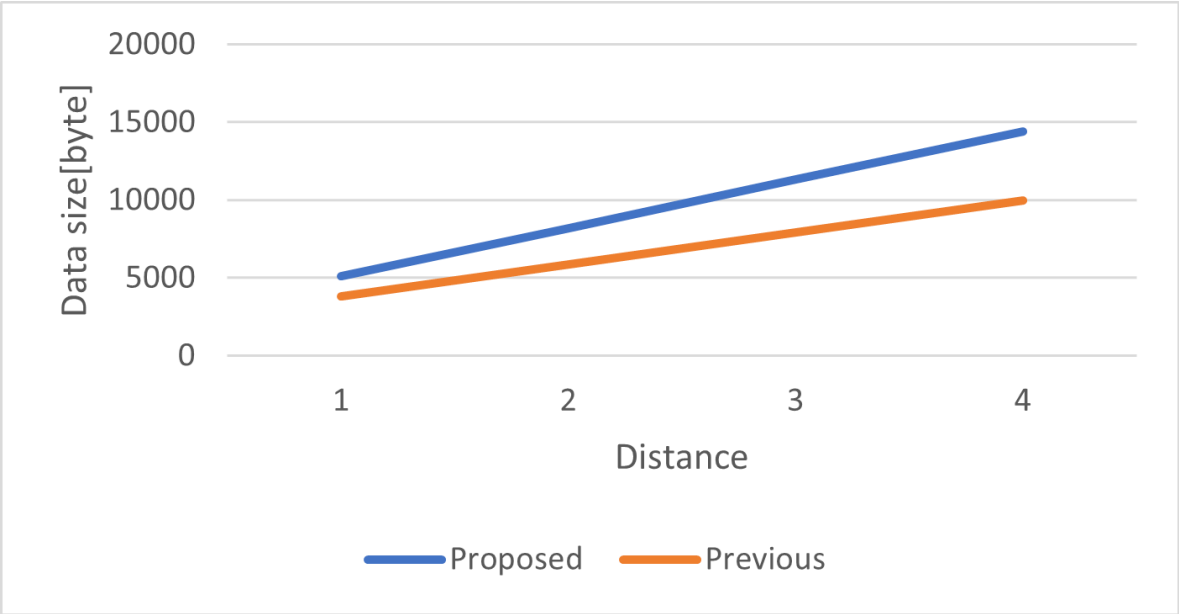


図7.8　証明する 2 点間の距離の変化による伝送データサイズの変化

証明する辺ごとに署名を伝送する必要があるため, 従来方式, 提案方式ともに距離に応じて線形で増加している. また, 従来方式に比べ提案方式ではラベルの証明のため 50% 程度データサイズが増加している.

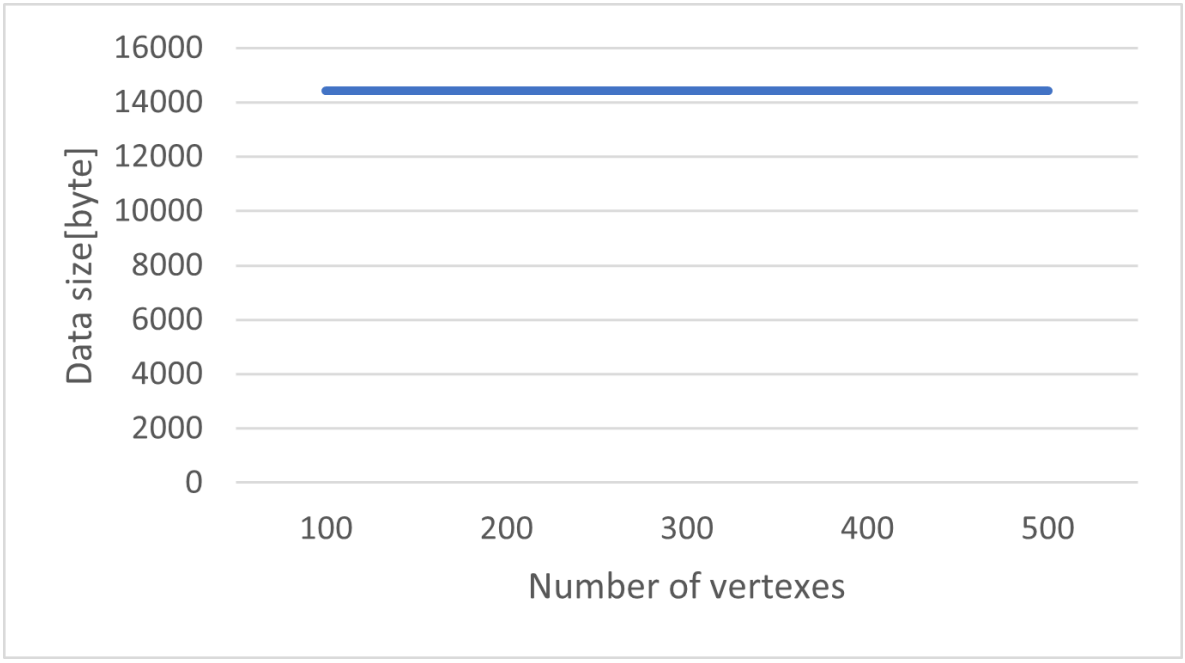


図7.9　点数の変化による伝送データサイズの変化

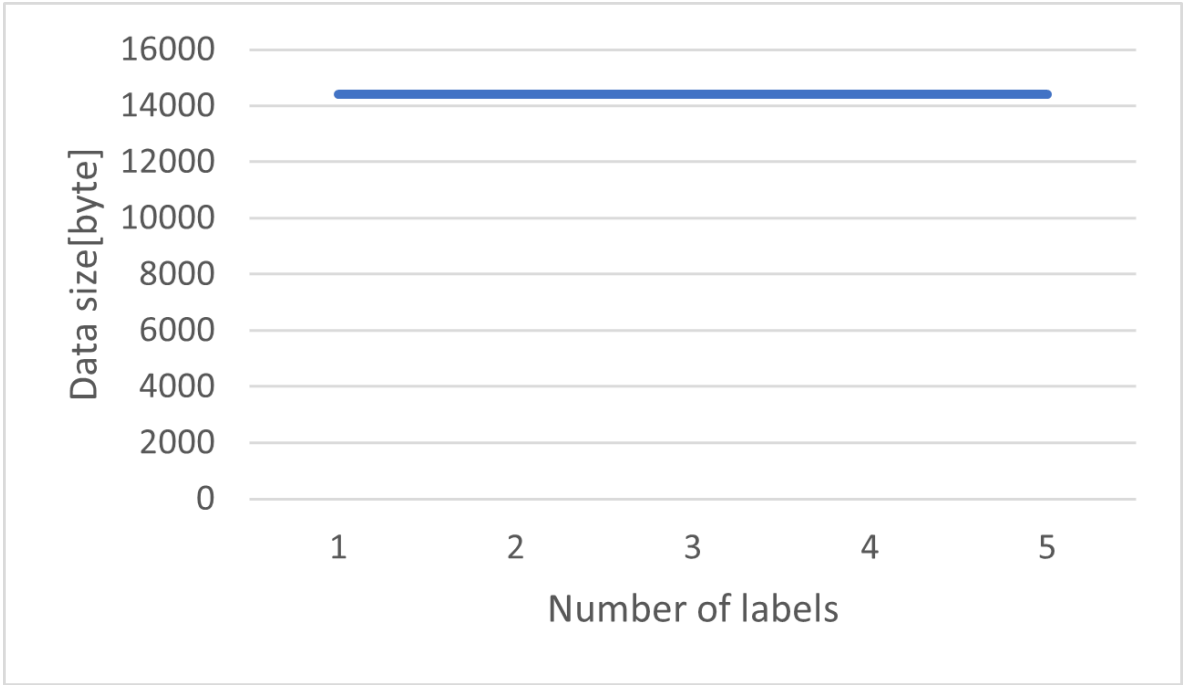


図7.10　ラベル数の変化による伝送データサイズの変化

点数, ラベル数についてはデータサイズは依存せず, 一定の値となっており, 15 キロバイト程度と実用上問題ない大きさであると言える.

# 第 8 章

# まとめ

本研究では, 辺がラベルを持つ有向グラフにおいて, 従来方式 [2] と同様にペアリングベースアキュームレータを用いた接続性のゼロ知識証明, および制約を持つ接続性のゼロ知識証明を提案した. 利点として, いずれも従来方式 [2] 同様, テナント側で行われる検証にかかる処理時間と証明データサイズが, 点数, 辺数, ラベル数に依存していない. しかし, その分従来方式 [2] に比べてオーバーヘッドが増加している. 今後の課題としては, 以下の 3 つが考えられる. 1 つ目に, プロバイダ側で行われる証明にかかる処理時間の内, 補助情報の計算時間が点数, 辺数, ラベル数に依存しているため, この部分を改善することである. 2 つ目に, 全ての処理時間が証明する 2 点間の距離に比例して大きくなるため, この部分を改善することである. 3 つ目に, 提案方式を利用した具体的なアプリケーションを実装することで, より実用的な実験を行うことである.

# 謝辞

本研究は広島大学大学院先進理工系科学研究科・計算機基礎学研究室において行ったものです. 本論文を作成するにあたり中西透教授, および研究室の大学院生から懇切丁寧かつ適切なご指導を賜りました. ここに謝意を表します. また, ゼミにおいて様々な知識, 助言を通してご助力いただきました北須賀輝明准教授, 岩本宙造教授, 福山大学の今井克暢准教授, またご協力頂きました学部生の皆様に感謝します.

# 参考文献


[1] T. Gross, “Efficient Certification and Zero-knowledge proofs of knowledge on infrastructure topology graphs,” Proc. 6th ACM Workshop on Cloud Computing Security (CCSW’14), pp69-80, 2014.

[2] T. Nakanishi, H. Yoshino, T. Murakami, and G. Policharla, “Efficient zero-knowledge proofs of graph signature for connectivity and isolation using bilinear-map accumulator,” Proc. 7th ASIA Public-Key Cryptography Workshop(APKC@AsiaCCS 2020), pp.9–18, 2020.

[3] D. Yamamoto, Y. Suga, and K. Sako, “Formalising linked-data based verifiable credentials for selective disclosure,” IEEE European Symposium on Security and Privacy Workshops (EuroS&PW), 2022.

[4] M. Abe, G. Fuchsbauer, J. Groth, K. Haralambiev, and M. Ohkubo, “Structure-preserving signatures and commitments to group elements,” Advances in Cryptology - CRYPTO 2010, LNCS 6223, pp.209-236, Springer-Verlag, 2010.

[5] M. Abe, K. Haralambiev, and M. Ohkubo, “Signing on elements in bilinear groups for modular protocol design,” Cryptology ePrint Archive, Report 2010/133, 2010.

[6] C. Papamanthou, R. Tamassia, and N. Triandopoulos, “Optimal verification of operations on dynamic sets,”Advances in Cryptology —CRYPTO 2011, LNCS 6841, pp.91–110, Springer–Verlag, 2011.

[7] Y. Takahashi, Y. Nanjo, T. Kusaka, Y. Nogami, T. Kanenari, T. Tatara, “An Implementation and Evaluation of Pairing Library ELiPS for BLS Curve with Several Techniques,” 34th International Technical Conference on Circuits/Systems, Computers and Communications (ITC-CSCC), 2019.