ペアリングベースアキュームレータを用いたグラフ情報の分離性のゼロ知識証明

村上 友樹[1,a)]　中西 透[1,b)]

概要：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できる．システム利用者であるテナントにとって重要なのは，自らのシステムがうまく接続されていること（接続性）と自らのシステムが他のテナントから分離されていること（分離性）である．ネットワークシステムを管理するプロバイダに対して，グラフで表現されたネットワークトポロジーを開示せずにテナントに正しい情報を開示する手法が求められている．解決策として，RSA ベースでグラフ情報の接続性，分離性をゼロ知識証明する方式が提案されている．その一方で，ペアリングベースアキュームレータを用いた方式が提案されており，生成する鍵データサイズが RSA ベース方式と比べて小さいことに加えて，証明データサイズ・検証時間がグラフの全点・全辺数に依存しないという利点がある．しかし，この方式では接続性のゼロ知識証明のみ行われおり，分離性のゼロ知識証明が行われていない．本研究では，このペアリングベースの従来方式を拡張して分離性のゼロ知識証明を行う方式を提案する．そして提案手法を PC 上で実装し，その処理時間を測定することにより評価する．

キーワード：ゼロ知識証明，グラフ，ペアリング，アキュームレータ

Zero-Knowledge Proof of Isolation in Graph Using Pairing-Based Accumulator

Tomoki Murakami[1,a)]　Toru Nakanishi[1,b)]

Abstract: Network systems can be represented using graphs. What is important for a tenant is that his system is well connected (connectivity) and that his system is separated from other tenants (isolation). There is a demand for a method of confirming the connectivity and isolation to tenants without disclosing the network topology. As a solution, an RSA-based zero-knowledge proof scheme for a graph has been proposed. On the other hand, a scheme using a pairing-based accumulator has been proposed, where the proof size and the verification time do not depend on the total number of all vertexes and edges. However, only the zero-knowledge proof of connectivity is equipped. In this research, we propose a zero-knowledge proof scheme for isolation by extending the previous pairing-based scheme. Then we implement the proposed scheme on a PC and evaluate it by measuring the processing time.

Keywords: zero-knowledge proof, graph, pairing, accumulator

1. 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)．このネットワークで接続されたシステムを利用するテナントにとって重要なのは，自らのシステムがうまく接続されていること（接続性）と自

[1] 広島大学大学院先進理工系科学研究科
Graduate School of Advanced Science and Engineering, Hiroshima University
[a)] m203472@hiroshima-u.ac.jp
[b)] t-nakanishi@hiroshima-u.ac.jp

らのシステムが他のテナントから分離されていること（分離性）である．よって，テナントはシステムを管理しているプロバイダへこれらの条件を満たしているか確認したいという欲求がある．しかし，ネットワークシステムを提供しているプロバイダは，グラフの全ての情報を開示することは機密上できない．さらに，グラフの情報だけを伝えてもテナントはその情報が正しいか判断できない問題もある．そのため，プロバイダはシステム利用者へグラフの 2 点間の接続性や分離性などの正当性を不必要な情報を開示することなく保証する手法を考える必要がある．

そこで，ゼロ知識証明を用いた手法が解決案として提案されている [1]．ゼロ知識証明とは，ある命題が正しいということを，それ以外の情報を教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全ての情報を開示することなく，グラフのどの点が接続され分離されているかを確認することができる．この時，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ため，その正しさが保証される．

従来方式 [1] では，RSA 暗号によりグラフ情報の接続性，分離性をゼロ知識証明する手法が提案されている．しかし，この方式では，ゼロ知識証明時にすべての点・辺数に依存してデータサイズが増加してしまう．その一方で，ペアリングベースのアキュ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，生成する鍵データサイズが RSA 暗号と比べて小さいという特徴がある．また，アキュームレータにより，ゼロ知識証明時にデータサイズ，検証時間がグラフ全体の点・辺数に依存しない．しかし，接続性のゼロ知識証明のみ行われており，分離性のゼロ知識証明が行われていない．

そこで本研究では，従来方式 [2] を拡張して分離性のゼロ知識証明を行う方式を提案する．そして提案手法を PC 上で実装し，その処理時間を測定することにより評価する．

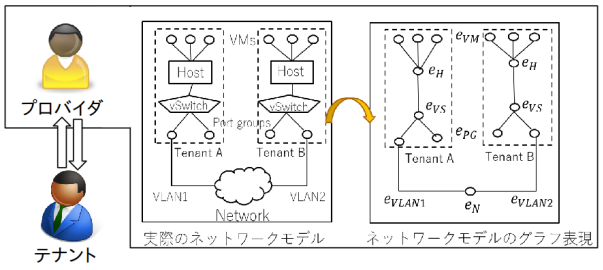

図 1　ネットワークモデル図

2. 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．G_1, G_2, G_T を素数の位数 p の巡回群とする．また，G_1，G_2 のそれぞれの生成元を g,$\tilde{g}$ とする．このとき，次に示す双線形写像 e:$G_1 \times G_2 \to G_T$ を定義できる．

双線形性： $P, P' \in G_1, Q, Q' \in G_2$, 任意の $a, b \in \mathbb{Z}_p$ に対し，
$e(PP', Q) = e(P, Q)e(P', Q)$,
$e(P, QQ') = e(P, Q)e(P, Q')$
及び $e(P^a, Q^b) = e(P, Q)^{ab}$ が成立する．

非退化性： $e(g, \tilde{g}) \neq 1_{G_T}$ (1_{G_T} は 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

定義 1. 双線形 q-strong Diffie-Hellman 仮定

k をセキュリティパラメータとし，$(p, G_1, G_2, G_T, e, g, \tilde{g})$ をランダムに生成された双線形ペアリングのパラメータの組とする．Z_p^* からランダムに選ばれた s を用いて表された $g, g^s, g^{s^2}, ..., g^{s^q} \in G_1, \tilde{g}, \tilde{g^s}, \tilde{g^{s^2}}, ..., \tilde{g^{s^q}} \in G_2$ が与えられた時，q が k を用いた多項式で表せるなら，無視できる確率 $neg(k)$ を除いて $a, e(g, \tilde{g})^{1/(s+a)}$ を出力できるような多項式時間アルゴリズムは存在しない．

定義 2. q-SFP 仮定

すべての多項式時間アルゴリズム A に対して，確率

$$\begin{aligned}\Pr\Big[\mathcal{A}\left(g_z, h_z, g_r, h_r, a, \tilde{a}, b, \tilde{b}, \{(z_j, r_j, s_j, t_j, u_j, v_j, w_j)\}_{j=1}^{q}\right) \\ = (z^*, r^*, s^*, t^*, u^*, v^*, w^*) \\ \wedge e(a, \tilde{a}) = e\left(g_z, z^*\right) e\left(g_r, r^*\right) e\left(s^*, t^*\right) \\ \wedge e(b, \tilde{b}) = e\left(h_z, z^*\right) e\left(h_r, u^*\right) e\left(v^*, w^*\right) \\ \wedge z^* \neq 1_{G_2} \wedge z^* \neq z_j \text{ for all } 1 \leq j \leq q\Big]\end{aligned}$$

は無視できる．ここで $\{(z_j, r_j, s_j, t_j, u_j, v_j, w_j)\}_{j=1}^{q}$ は

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

を満たす．また 1_{G_2} は G_2 の単位元である．

2.3 知識の証明

知識の証明とは, 証明者と検証者の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ことを秘密情報を開示することなく証明することができる. 本研究では, 離散対数である秘密情報 x を知っていることを示す知識の証明を用いる．$y = g_1^{x_1} g_2^{x_2} ...$ のような複数の底に対する離散対数の証明も可能である. そして, 同じ秘密情報を含むような複数の関係式も証明できる.

2.4 AHO 署名

AHO 署名 [3] とは複数の群要素の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．n 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：

まず署名するメッセージの数 n を与え，$G_r, H_r \in G_1$, $\tilde{g} \in G_2, \alpha_a, \alpha_b, \gamma_z, \delta_z, (\gamma_i, \delta_i)_{i=1}^n \in Z_p$ をそれぞれランダムに選ぶ. 次に $G_z = G_r^{\gamma}, H_z = H_r^{\delta}, G_i = G^{\gamma_i} H_i = H^{\delta_i}, A = e(G_r, \tilde{g}^{\alpha_a})$ と $B = e(H_r, \tilde{g}^{\alpha_b})$ を計算する．

AHO 署名の公開鍵
$pk_{AHO} = (p, e, \tilde{g}, G_r, H_r, G_z, H_z, (G_i, H_i)_{i=1}^n)$
AHO 署名の秘密鍵
$sk_{AHO} = (\alpha_a, \alpha_b, \gamma_z, \delta_z, (\gamma_i, \delta_i)_{i=1}^n)$
を出力する．

AHO Sign：

秘密鍵 $sk_{AHO} = (\alpha_a, \alpha_b, \gamma_z, \delta_z, (\gamma_i, \delta_i)_{i=1}^n)$ を用いてメッセージ $(M_1, ..., M_n) \in {G_2}^n$ に署名するために，ランダムに $\mu, \rho_a, \rho_b, \omega_a, \omega_b \in Z_p$ を選択し，以下のように計算して署名を作成する．

$\theta_1 = \tilde{g}^{\mu}, \theta_2 = \tilde{g}^{\rho_a - \gamma_z \mu} \cdot \prod_{i=1}^n M_i^{-\gamma_i}, \theta_3 = G_r^{\omega_a},$
$\theta_4 = \tilde{g}^{(\alpha_a - \rho_a)/\omega_a}, \theta_5 = \tilde{g}^{\rho_b - \delta_z \mu} \cdot \prod_{i=1}^n M_i^{-\delta_i},$
$\theta_6 = H_r^{\omega_b}, \theta_7 = \tilde{g}^{(\alpha_b - \rho_b)/\omega_b}$

そして署名を $\sigma = (\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6, \theta_7)$ とする．

AHO Verify：

署名 $\sigma = (\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6, \theta_7)$ が以下の検証式を満たしているなら受理する．

$A = e(G_z, \theta_1) \cdot e(G_r, \theta_2) \cdot e(\theta_3, \theta_4) \cdot \prod_{i=1}^n e(G_i, M_i)$

$B = e(H_z, \theta_1) \cdot e(H_r, \theta_5) \cdot e(\theta_6, \theta_7) \cdot \prod_{i=1}^n e(H_i, M_i)$

AHO 署名は, q-SFP 仮定に基づき, existential unforgeability が証明されている. 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ムレータ [4] は，ある要素が集合に含まれているかを効率良く証明できる手法である．Z_p^* の要素から成る集合 $\mathcal{X}$ に対して, ランダムに選ばれた要素 $s \in Z_p^*$ を用意し,$\mathcal{X}$ アキュムレートされた値 $acc(\mathcal{X})$ を $acc(\mathcal{X}) = g^{\prod_{x \in \mathcal{X}}(x+s)}$ と定義する．また $\prod_{x \in \mathcal{X}}(x+s)$ を展開すると $a_0 + a_1 s + a_2 s^2 + ...$ のように s の多項式になるため，$g^s, g^{s^2}, ...$ を用いて s なしで $acc(\mathcal{X})$ を計算することができる．

集合 $S \subseteq \mathcal{X}$ に対して部分集合包含の証明には以下で定義された証拠情報 $W_{S,\mathcal{X}}$ を用いる．

$$W_{S,\mathcal{X}} = g^{\prod_{x \in \mathcal{X}-S}(x+s)}$$

検証は $e(W_{S,\mathcal{X}}, \tilde{g}^{\prod_{x \in S}(x+s)}) = e(acc(\mathcal{X}), g)$ を満たしているかどうかで行う．このアキュムレータは双線形 q-SDH 仮定に基づき安全である．

3. 先行研究の概要

本章では, 先行研究として, ペアリングベースアキュームレータを用いた，グラフに対する接続性のゼロ知識証明 [2] の概要を示す.

この方式では，従来方式 [1] と同様にグラフの各ノードに頂点識別子として素数 x_i を割り当て，辺情報は両端の頂点識別子を掛け合わせた $x_i x_j$ と表現する． そして，点集合 V と辺集合 E のグラフの署名にはペアリングベースのゼロ知識証明に適した AHO 署名を用いる．各点，各辺をペアリングベースのアキュムレータで acc_V，acc_E に圧縮し，それをメッセージとして署名する．acc_V，acc_E をコミットメントして AHO 署名の証明をすることにより，グラフの情報を秘匿してその正当性をゼロ知識証明できる．この時，隣接している 2 点間の接続性の証明は，両端の点の値が辺の値を割り切ることを証明することで可能となる．これを証明したい 2 点間の全ての辺で行うことにより，2 点間の接続が示される．

この方式では，すべての点・辺情報をアキュームレータで圧縮した後で署名される．署名の検証が単一のアキュムレータに対してのみ行え，アキュームレータに圧縮されていることを単一のペアリング式で証明できるため，全点・辺数にデータサイズ，検証時間が依存しないことが利点である．一方，分離性の証明は考慮されてない．

4. 提案方式

4.1 提案方式の概要

提案方式では, 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と AHO 署名を用いて分離性のゼロ知識証明を行う. まず, 対象のグラフに対して分離された部分グラフ毎にブロックとして分割する. 各ブロック内の点は連結であり，異なるブロックの任意の 2 点は非連結とする．そ

して, 各ブロックごとの点情報をアキュームレータ acc_{Vk} に圧縮する. 点情報は，辺が割り切るかを示す必要がないため素数に限定する必要はない．次に，生成したアキュームレータ acc_{Vk} をメッセージとして AHO 署名を行い，グラフ情報の正しさを保証する．次に 2 点 i, j 間の分離を示すには，$i \in V_{k_i}, j \in V_{k_j}$ であることをアキュームレータによる包含関係により示す．そして，$acc_{V_{k_i}}, acc_{V_{k_j}}$ が署名された acc_{V_k} のいずれかであることを k を明らかにすることなく示すとともに，$acc_{V_{k_i}} \neq acc_{V_{k_j}}$ を示すことにより，2 点が同じブロックに存在しないことを示す．

RSA ベースの従来方式 [1] では，Integer Commitment を利用することで，互いに素の整数上の関係式を利用して非連結性を証明している．しかし，ペアリングベースでは整数上の関係式を証明できないため，このような手法を採用している．

4.2 提案プロトコル

4.2.1 グラフ情報の署名

アキュームレータの構成

各点 i に $x_i \in Z_p^*$ を割り当てる．グラフを K 個のブロックに分割した場合，各ブロックを $V_1, ...V_K$ とする．それぞれのブロックの点情報のアキュームレータは以下となる．

$$acc_{V_k} = g^{\prod_{i \in V_k}(x_i+s)} \tag{1}$$

AHO 署名

それぞれのブロックの点情報に署名するために，acc_{V_k} $(k = 1, .., K)$ をメッセージとして AHO 署名を作成する．この時，以下により AHO 署名の検証ができる．

$$\begin{aligned} A &= e(G_z, \theta_1) \cdot e(G_r, \theta_2) \cdot e(\theta_3, \theta_4) \\ &\quad \cdot e(G_1, acc_{V_1}) \cdots e(G_K, acc_{V_K}) \end{aligned} \tag{2}$$

$$\begin{aligned} B &= e(H_z, \theta_1) \cdot e(H_r, \theta_5) \cdot e(\theta_6, \theta_7) \\ &\quad \cdot e(H_1, acc_{V_1}) \cdots e(H_K, acc_{V_K}) \end{aligned} \tag{3}$$

4.2.2 2 点間の分離性のゼロ知識証明

グラフ中の 2 点 i, j 間の分離を示す際，グラフ全体の署名の知識を証明し，その後，i, j の異なるブロックへの存在を証明し, 2 点間が分離していることを証明する．

グラフ情報の署名の証明

(1) コミットメントの作成

グラフ情報 (V, E) の AHO 署名 $(\theta_1, ..., \theta_7)$ をランダマイズして $(\theta'_1, ..., \theta'_7)$ を得る．$r_{\theta_1}, r_{\theta_2}, r_{\theta_5} \in Z_p*$ を選び，$\theta'_1, \theta'_2, \theta'_5$ を以下のようにコミットメントする．

$$\begin{aligned} C_{\theta_1} &= \theta'_1 \cdot S^{r_{\theta_1}} \\ C_{\theta_2} &= \theta'_2 \cdot S^{r_{\theta_2}} \\ C_{\theta_5} &= \theta'_5 \cdot S^{r_{\theta_5}} \end{aligned}$$

AHO 署名されたメッセージである $acc_{V_1}, \cdots, acc_{V_K}$ に対して，$r_{acc_{V_1}}, ..., r_{acc_{V_K}} \in Z_p$ をランダムに選び，以下のようにコミットメントを生成する．

$$C_{acc_{V_1}} = acc_{V_1} \cdot S^{r_{acc_{V_1}}} \tag{4}$$

$$\cdots$$

$$C_{acc_{V_k}} = acc_{V_k} \cdot S^{r_{acc_{V_k}}} \tag{5}$$

$$\cdots$$

$$C_{acc_{V_K}} = acc_{V_K} \cdot S^{r_{acc_{V_K}}} \tag{6}$$

(2) AHO 署名の知識の証明

AHO 署名の検証式に上記のコミットメントを代入して以下の式を得る．

$$\begin{aligned} &A \cdot e(G_z, C_{\theta_1})^{-1} \cdot e(G_r, C_{\theta_2})^{-1} \cdot e(\theta'_3, \theta'_4)^{-1} \\ &\cdot \{e(G_1, C_{acc_{V_1}})\}^{-1} \cdots \{e(G_K, C_{acc_{V_K}})\}^{-1} \\ &= e(G_z, S^{-1})^{r_{\theta_1}} \cdot e(G_r, S^{-1})^{r_{\theta_2}} \\ &\cdot e(G_1, S^{-1})^{r_{acc_{V_1}}} \cdots e(G_2, S^{-1})^{r_{acc_{V_K}}} \end{aligned} \tag{7}$$

$$\begin{aligned} &B \cdot e(H_z, C_{\theta_1})^{-1} \cdot e(H_r, C_{\theta_5})^{-1} \cdot e(\theta'_6, \theta'_7)^{-1} \\ &\cdot \{e(G_1, C_{acc_{V_1}})\}^{-1} \cdots \{e(G_2, C_{acc_{V_K}})\}^{-1} \\ &= e(H_z, S^{-1})^{r_{\theta_1}} \cdot e(H_r, S^{-1})^{r_{\theta_5}} \\ &\cdot e(H_1, S^{-1})^{r_{acc_{V_1}}} \cdots e(H_K, S^{-1})^{r_{acc_{V_K}}} \end{aligned} \tag{8}$$

この 2 つの式に対して，知識の証明を行う．

2 点間の分離の証明

(1) 点 i, j の存在証明

点 i と点 j が，ブロック $k_i, k_j \in [1, K]$ に存在することを，アキュームレータの検証により証明する．
そのために，補助情報を以下のように導出する．

$$W_i = g^{\prod_{t \in V_{k_i}, t \neq i}(x_t+s)} \tag{9}$$

$$W_j = g^{\prod_{t \in V_{k_j}, t \neq j}(x_t+s)} \tag{10}$$

以下のように W_i, W_j のコミットメントを生成する．

$$C_{W_i} = W_i \cdot S^{r_{W_i}} \tag{11}$$

$$C_{W_j} = W_j \cdot S^{r_{W_j}} \tag{12}$$

アキュームレータ $acc_{V_{k_i}}$ と $acc_{V_{k_j}}$ のコミットメントを以下のように計算する．

$$C_{acc_{V_{k_i}}} = acc_{V_{k_i}} \cdot S^{r_{acc_{V_{k_i}}}} \tag{13}$$

$$C_{acc_{V_{k_j}}} = acc_{V_{k_j}} \cdot S^{r_{acc_{V_{k_j}}}} \tag{14}$$

コミットメント $C_{acc_{V_{k_i}}}, C_{acc_{V_{k_j}}}, C_{W_i}, C_{W_j}$ に対してアキュームレータの検証式は以下となる．

$$\begin{aligned} &e(C_{W_i}, g^s) \cdot e(C_{W_i}, g^{x_i}) \cdot e(C_{acc_{V_{k_i}}}, g)^{-1} \\ &= e(S, g^{x_i})^{r_{W_i}} \cdot e(S, g^s)^{r_{W_i}} \cdot e(S^{-1}, g)^{r_{acc_{V_{k_i}}}} \end{aligned} \tag{15}$$

$$\begin{aligned} &e(C_{W_j}, g^s) \cdot e(C_{W_j}, g^{x_j}) \cdot e(C_{acc_{V_{k_j}}}, g)^{-1} \\ &= e(S, g^{x_j})^{r_{W_j}} \cdot e(S, g^s)^{r_{W_j}} \cdot e(S^{-1}, g)^{r_{acc_{V_{k_j}}}} \end{aligned} \tag{16}$$

この式に対し，r_{W_i},$r_{acc_{V_{k_i}}}$,r_{W_j},$r_{acc_{V_{k_j}}}$ を秘密情報として知識の証明を行う．この式は以下のアキュームレータの検証式にコミットメントを代入して整理したものである．

$$e(W_i, g^{(x_i+s)}) = e(acc_{V_{k_i}}, g) \tag{17}$$

$$e(W_j, g^{(x_j+s)}) = e(acc_{V_{k_j}}, g) \tag{18}$$

(2) V_{k_i} および V_{k_j} の V_k に対する OR 証明

acc_{V_k} $(1 \leq k \leq K)$ のいずれかが $acc_{V_{k_i}}$ であること，acc_{V_k} $(1 \leq k \leq K)$ のいずれかが $acc_{V_{k_j}}$ であることを示す．$acc_{V_k} = acc_{V_{k_i}}$ より，コミットメントの式 (5) が式 (13) で割り切れば良い．つまり，以下となる．

$$C_{acc_{V_k}} \cdot {C_{acc_{V_{k_i}}}}^{-1} = S^{r_{acc_{V_k}} - r_{acc_{V_{k_i}}}} \tag{19}$$

$1 \leq k \leq K$ のうち 1 つ k で成立すれば良いので，論理和 OR を使って次のように表せる．

$$\begin{aligned} & C_{acc_{V_1}} \cdot {C_{acc_{V_{k_i}}}}^{-1} = S^{r_{acc_{V_1}} - r_{acc_{V_{k_i}}}} \\ \vee & C_{acc_{V_2}} \cdot {C_{acc_{V_{k_i}}}}^{-1} = S^{r_{acc_{V_2}} - r_{acc_{V_{k_i}}}} \\ & \vee \cdots \\ \vee & C_{acc_{V_K}} \cdot {C_{acc_{V_{k_i}}}}^{-1} = S^{r_{acc_{K_2}} - r_{acc_{V_{k_i}}}} \end{aligned}$$

この関係式を OR 型の知識の証明 [5] を用いて示す．点 j についても同様である．

(3) $V_{k_i} \neq V_{k_j}$ の証明

(3-1) 同じブロックに点 i と点 j の両方が含まれていないことを示す．$acc_{V_{k_i}} \neq acc_{V_{k_j}}$ なので，式 (13) の両辺を式 (14) で割り R 乗することで，以下の式が得られる．

$$\left(\frac{acc_{V_{k_i}}}{acc_{V_{k_j}}}\right)^R = (C_{acc_{V_{k_i}}})^R \cdot ({C_{acc_{V_{k_j}}}}^{-1})^R \cdot (S^{-1})^{{r_{acc_{V_{k_i}}}}^R} \cdot S^{{r_{acc_{V_{k_j}}}}^R} \tag{20}$$

$(\frac{acc_{V_{k_i}}}{acc_{V_{k_j}}})^R$ を $\tilde{C}$ と置く．$acc_{V_{k_i}} \neq acc_{V_{k_j}}$ より，$\tilde{C} \neq 1$ である．式 (20) より以下の式を得る．

$$\tilde{C} = (C_{acc_{V_{k_i}}} \cdot {C_{acc_{V_{k_j}}}}^{-1})^R \cdot (S^{-1})^{{r_{acc_{V_{k_i}}}}^R} \cdot S^{{r_{acc_{V_{k_j}}}}^R} \tag{21}$$

この式に対し，$\alpha = r_{acc_{V_{k_i}}} \cdot R$，$\beta = r_{acc_{V_{k_j}}} \cdot R$ を秘密情報として知識の証明を行う．

(3-2) $\alpha = r_{acc_{V_{k_i}}} \cdot R$ を証明する．($\beta = r_{acc_{V_{k_j}}} \cdot R$ も同様である)．

R と α に対するコミットメントを以下のように作成する．

$$C_R = (L)^R S^{\rho_1} \tag{22}$$

$$C_\alpha = (L)^\alpha S^{\rho_2} \tag{23}$$

そして上記の式と以下に対して知識の証明を行う．

$$C_\alpha = {C_R}^{r_{acc_{V_{k_i}}}} S^{\rho'} \tag{24}$$

4.3 安全性

4.3.1 健全性

i，j 間の分離性の証明において，監査人の AHO 署名の検証により各 acc_{V_k} は正しいことが保証される．また，アキュームレータの検証により i, j は V_{k_i}, V_{k_j} に存在するとともに，OR 証明により V_{k_i}, V_{k_j} は V_k のいずれかであることが保証される．さらに，$V_{k_i} \neq V_{k_j}$ が証明されていることから，i, j は異なるブロックであることが示されている．

4.3.2 ゼロ知識性

コミットメントおよび知識の証明からは，証明していること以外の情報は洩れない．OR 型の証明では V_{k_i}, V_{k_j} が V_k のいずれかであるか分からない．$V_{k_i} \neq V_{k_J}$ の証明では，R 乗することにより，どの V_{k_i}, V_{k_j} か分からないようになっている．

5. 実装・実験結果

5.1 実装環境

表 1 の環境で提案方式の実装を行い，分離性のゼロ知識証明の処理時間を計測した．

表 1 実装環境

OS	Ubuntu 16.04
CPU	Intel(R)Core(TM)i5-9400(2.90GHz)
プログラミング言語	C 言語
多倍長ライブラリ	GMP 6.0.0
ペアリングライブラリ	ELiPS[6]

5.2 実装結果

まず，グラフのブロック数を 5 に固定し，ブロック内の点数を変更して，全体の点数が 500 になるまで処理時間を調べた．図 2 から分かるように，検証時間はほぼ一定であるが，証明時間は増加している．次に，グラフのブロック数を 10 に固定し，ブロック内の点数を変更して，全体の点数が 1000 になるまで処理時間を調べた．こちらも，図 3 から分かるように，検証時間はほぼ一定であるが，証明時間は増加している．また，2 つのグラフから点数が増えても全体の処理時間がそれほど増えていないことがわかる．証明時間が増加しているのは，補助情報 W_i, W_j の計算時間が，ブロック内の点数 n に依存しているためである．

次に，ブロック内の点数を 10 に固定し，ブロック数を変更して，全体の点数が 500 になるまで処理時間を調べた．図 4 から分かるように，証明時間，検証時間ともに増加しており，全体の処理時間が大きく増加してしまう．さらにブロック内の点数を 20 に固定し，ブロック数を変更して，全体の点数が 1000 になるまで処理時間を調べた．図 5 から分かるように，証明・検証時間ともに増加している．これはグラフ証明の証明および分離の証明における (2) の OR 証明がブロック数 K に依存しているためである．

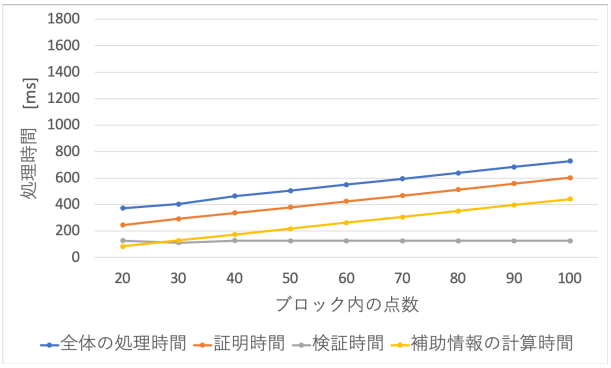

図 2 ブロック内の点数に対する処理時間の変化 (ブロック数 5)

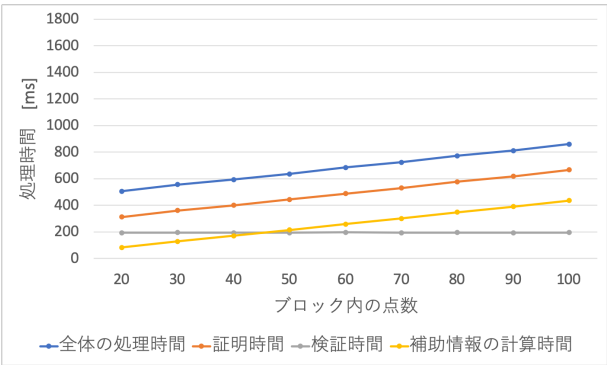

図 3 ブロック内の点数に対する処理時間の変化 (ブロック数 10)

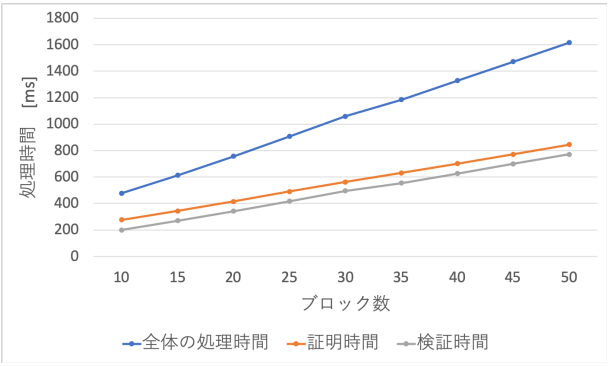

図 4 ブロック数に対する処理時間の変化 (ブロック内点数 10)

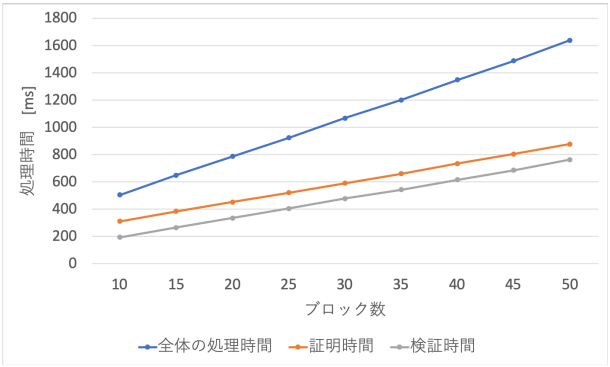

図 5 ブロック数に対する処理時間の変化 (ブロック内点数 20)

6. まとめ

本研究では，従来方式 [2] を拡張して，ペアリングアキュームレータを用いたグラフの分離性のゼロ知識証明を提案した．ブロック数が小さい場合においては，処理時間が実用的であることを確認した．しかし，ブロック数が大きければ大きいほど処理時間が増えてしまい問題となりうることが分かった．

今後の課題として，ブロック数に依存する証明・検証時間を小さくすることが挙げられる．

参考文献

参考文献

[1] T. Gross, “Efficient Certification and Zero-Knowledge Proofs of Knowledge on Infrastructure Topology Graphs”, CCSW ’14, pp.69-80, 2014.

[2] 吉野大海, 中西透, “ペアリングを用いたグラフ情報のゼロ知識証明の高速化”, 信学技報, vol. 119, no. 257, ISEC2019-67, pp. 33-39, 2019 年 11 月.

[3] M. Abe, K. Haralambiev, M. Ohkubo, “Signing on Elements in Bilinear Groups for Modular Protocol Design”, Cryptology ePrint Archive, 2010/133.

[4] C. Papamanthou, R. Tamassia and N. Triandopoulos, “Optimal Vertification of Operationd on Dynamic Sets”, Advances in Cryptology CRYPTO 2011, LNCS6841, pp.91-110, 2011.

[5] R. Cramer, I. Damgard, B. Schoenmakers, “Proofs of Partial Knowledge and Simplified Design of Witness Hiding Protocols”, CRYPTO 1994, LNCS 839, pp.174-187, 1994.

[6] 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.