Computer Security Symposium 2020
26 - 29 October 2020

CSベース失効可能グループ署名における ベクトルコミットメントを用いた失効リストの削減

杉本　光平[1,a)]　中西　透[1,b)]

概要：現在用いられている ID ベースのユーザ認証では, ユーザ ID が紐付けできるため, ユーザ ID からユーザの利用履歴を収集できてしまう. そのため, 利用履歴の漏洩などのプライバシー問題が発生する可能性がある. その問題点の解決策としてグループ署名が研究されている. グループ署名では, グループから脱退, もしくは削除されたユーザがその後に生成した署名を正しくない署名として処理する失効機能が必要である. 効率的な失効可能グループ署名として，CS(Complete Subtree) 法と呼ばれる木構造を用いた手法に基づいた方式が提案されている. しかしこの方式では, 失効リストのサイズが失効ユーザ数に対して大きく依存しているという問題点がある. そこで本研究では, 失効リストの各情報をブロックに分割しベクトルコミットメントで圧縮を行うことにより, 失効リストのサイズを削減した方式を提案する. さらに PC 上において実装し, 提案方式の有用性を評価する.

キーワード：グループ署名，失効，プライバシー保護

Reducing Revocation Lists in CS-Based Revocable Group Signature Scheme Using Vector Commitment

Kouhei Sugimoto[1,a)]　Toru Nakanishi[1,b)]

Abstract: In the current ID-based user authentication systems, the user's personal information of use history can be traced, because the user's ID can be linked. Therefore, this may cause privacy problems for the users. Group signatures have been researched as a solution to this problem. The group signature requires a revocation function to revoke signatures generated by a user who leaves or is removed from a group. An efficient revocable group signature scheme using a tree structure called the Complete Subtree (CS) method has been proposed. However, the size of the revocation list depends on the number of revoked users. In this paper, we propose a new scheme to reduce the size of revocation list by dividing each information in the list into blocks and compressing them with vector commitment. We implement this method on a PC, and evaluate the effectiveness of the proposed scheme.

Keywords: Group signature，Revocation，Privacy protection

1. はじめに

現在広く利用されている ID に基づいたユーザ認証では, ユーザ ID と氏名, 住所, 電話番号などの個人情報やサービスの利用履歴といった情報が関連付けられている. このユーザと紐付けられた情報はサーバに蓄積される. しかし, サーバに蓄積された情報が漏洩してしまう恐れがあり, ユーザのプライバシー問題となり得る. この問題の解決策としてグループ署名 [1] という認証方式が提案されている. グループ署名とは, あるグループに所属しているユーザが「グループに所属している」ということを匿名で証明できる署名方式である. グループ署名により, 匿名で正規のユーザかどうかについて確認できるため, ユーザ ID がサーバに渡るこ

[1] 広島大学
Hiroshima University
[a)] m202228@hiroshima-u.ac.jp
[b)] t-nakanishi@hiroshima-u.ac.jp

とがなく, プライバシーを保護したユーザ認証が可能となる. グループ署名では GM(Group Manager) と呼ばれるグループ管理者が, メンバーのグループへの加入を管理する. さらに，グループから脱退, もしくは不正を行い削除されたユーザが，その後に生成した署名を匿名性を維持したまま正しくない署名として処理する失効機能も必要であり，失効可能グループ署名が数多く提案されている. この失効機能では, 失効リスト (Revocation List:RL) と呼ばれる失効されたユーザの情報が記されたリストを利用する.

効率的な失効可能グループ署名として [2] が提案されている. [2] では, 失効リストの生成において CS 法を用いた方式と SD 法を用いた方式の 2 つの方式を提案しており，N を全ユーザ数とした時, 署名者に関するコストが最大でも $\mathcal{O}(\log N)$ である. しかし, この方式では standard モデルで構築され, Groth-Sahai 証明を使用しているため， 署名長が RSA 署名の署名長と比べて，160-bit Security の場合において約 10 倍大きくなる. そこで [2] の方式を改良した [3] が提案されている. この方式ではランダムオラクルを用いており，[2] で使用された CS 法と呼ばれる木構造でのユーザ管理と BBS+署名および Fiat-Shamir 変換した Schnorr ベースのゼロ知識証明（SPK）を用いることにより, r を失効ユーザ数, N を全ユーザ数とした時, 定数オーダーの署名・検証コスト, 高々 $\mathcal{O}(\log N)$ の鍵サイズを維持しつつ署名サイズを軽減している. しかし, 失効リストのサイズが $\mathcal{O}(r \log \frac{N}{r})$ となってしまい, 失効ユーザ数に大きく依存してしまう.

一方, [4] では, ベクトルコミットメントと呼ばれる, ベクトルの要素を圧縮してコミットする手法を用いて, [2] で使用された SD 法と呼ばれる木構造から生成される失効リスト中の失効情報を圧縮パラメータ T 個ごとに圧縮して署名することで, 失効リストのサイズを $\mathcal{O}(\frac{r}{T})$ に削減している. しかし, ベクトルコミットメントの検証におけるゼロ知識証明の処理が増えており, その計算の分のオーバヘッドが生じている. また，[2] と同様に Groth-Sahai 証明を用いており，署名長が大きい.

本研究では，ベクトルコミットメントを用いることにより，短い署名長の従来方式 [3] の失効リストを削減することを目的とする. さらに PC 実装において提案方式の有効性を検討する.

2. 数学的準備

2.1 双線形写像

本研究では以下の双線形群を利用する。$\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T$ を位数 p の巡回群とする。$\mathbb{G}_1, \mathbb{G}_2$ の生成元をそれぞれ g, h とする。このとき双線形写像 $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ は以下の双線形性と非退化性を満たす。

- **双線形性：**任意の $P, Q \in \mathbb{G}_1, R, S \in \mathbb{G}_2, a, b \in \mathbb{Z}_p$ に対して

  $e(PQ, R) = e(P, R)e(Q, R)$
  $e(P, RS) = e(P, R)e(P, S)$
  $e(P^a, R^b) = e(P, R)^{ab}$
- **非退化性：**$e(g, h) \neq 1$

上記の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

- **Definition 1 (q-SDH 仮定)**
  全ての確率的多項式時間 (PPT) アルゴリズム $\mathcal{A}$ において, 以下の確率

  $$Pr\left[\mathcal{A}(g, g^{\gamma^i}(i = 0, \cdots, q), h, h^{\gamma}) = (c, g^{\frac{1}{(\gamma+c)}} \wedge c \in \mathbb{Z}_p)\right]$$

  (ただし $g \in_R \mathbb{G}_1, \gamma \in_R \mathbb{Z}_p$) は無視できる.
- **Definition 2 (n-DHE 仮定)**
  全ての確率的多項式時間 (PPT) アルゴリズム $\mathcal{A}$ において, 以下の確率

  $$Pr\left[\mathcal{A}(g, g^a, \ldots, g^{a^n}, g^{a^{n+2}}, \ldots, g^{a^{2n}}) = g^{a^{n+1}}\right]$$

  (ただし $g \in_R \mathbb{G}, a \in_R \mathbb{Z}_p$) は無視できる.
- **Definition 3 (q-SFP 仮定)**
  全ての確率的多項式時間 (PPT) アルゴリズム $\mathcal{A}$ において, 以下の確率

  $$Pr\left[\mathcal{A}(g_z, h_z, g_r, h_r, a, \bar{a}, b, \bar{b}, \{(z_j, r_j, s_j, t_j, u_j, v_j, w_j)\}_{j=1}^{q}\right.$$
  $$= (z^*, r^*, s^*, t^*, u^*, v^*, w^*)$$
  $$\wedge e(a, \bar{a}) = e(g_z, z^*)e(g_r, r^*)e(s^*, t^*)$$
  $$\wedge e(b, \bar{b}) = e(h_z, z^*)e(h_r, u^*)e(v^*, w^*)$$
  $$\left.\wedge z^* \neq 1_{\mathbb{G}_2} \wedge z^* \neq z_j (1 \leq j \leq q)\right]$$

  は無視できる. ただし $(a, b, g_z, h_z, g_r, h_r) \in \mathbb{G}_1^6, (\bar{a}, \bar{b}) \in \mathbb{G}_2^2, (s_j, v_j)_{j=1}^q \in \mathbb{G}_1^2, (z_j, r_j, t_j, u_j, w_j)_{j=1}^q \in \mathbb{G}_2^5$ とし，以下の式を満たす.
  $e(a, \bar{a}) = e(g_z, z_j)e(g_r, r_j)e(s_j, t_j)$
  $\wedge e(b, \bar{b}) = e(h_z, z_j)e(h_r, u_j)e(v_j, w_j)$

2.3 知識の署名

知識の署名 (SPK : Signature based on Proof of Knowledge) は知識のゼロ知識証明を変換することで得られる。知識のゼロ知識証明とは証明者 P と検証者 V 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漏らすことなく証明する。離散対数の秘密情報 x を知ることを示すメッセージ m における SPK は以下のように記述される。

$$SPK\{(x) : y = g^x\}(m)$$

2.4 BBS+署名

従来方式 [3] と同様に，q-SDH 仮定に基づいた BBS+署

名 [6], [7] を用いる．BBS+署名のアルゴリズムは以下の通りである. 事前にセットアップとして, $g, g_1, \cdots, g_L, g_{L+1}$ を $\mathbb{G}_1$ から，h を $\mathbb{G}_2$ からランダムに生成する.

- **KeyGen**
  まず，γ を $\mathbb{Z}_p$ からランダムに選び, $\omega = h^\gamma$ とする. そして, 公開鍵を $vk = \omega$, 秘密鍵を $sk = \gamma$ とする.

- **Sign**
  メッセージを $(m_1, \cdots, m_L) \in \mathbb{Z}_p$ とする. まず, η, ζ を $\mathbb{Z}_p$ からランダムに選び，$A = (g_0 g_1^{\zeta} g_2^{m_1} \cdots g_{L+1}^{m_L})^{\frac{1}{\eta+\gamma}}$ を計算する. 署名は $\sigma = (A, \eta, \zeta)$ とする.

- **Verify**
  署名 $\sigma = (A, \eta, \zeta)$ とメッセージを $(m_1, \cdots, m_L)$ に対して, $e(A, h^\eta vk) = e(g_0 g_1^{\zeta} g_2^{m_1} \cdots g_{L+1}^{m_L}, h)$ を満たす場合は (valid) が出力され, そうでなければ (invalid) が出力される.

2.5 AHO 署名

AHO 署名 [5] は複数の群要素のメッセージに署名できる方式である。また署名検証のペアリングの関係式をゼロ知識証明できる。本研究では，群要素であるベクトルコミットに署名してそれをゼロ知識証明するため，AHO 署名を用いる。AHO 署名は q-SFP 仮定の基で安全である。
この AHO 署名はそれぞれ下記の 3 つのプロトコルで構成される。

- **AHOkeyGen**
  双線形群のパラメータ $(p, e, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T)$ を選び，$G_r, H_r \in_R \mathbb{G}_2$, $g \in_R \mathbb{G}_1$, $\alpha_a, \alpha_b, \gamma_z, \delta_z, (\gamma_i, \delta_i)_{i=1}^n \in_R \mathbb{Z}_p$ を選ぶ．次に $G_z = G_r^{\gamma}, H_z = H_r^{\delta}, G_i = G^{\gamma_i}, H_i = H^{\delta_i}, A = e(G_r, g^{\alpha_a})$ と $B = e(H_r, g^{\alpha_b})$ を計算する．

  AHO 署名の公開鍵
  $pk_{AHO} = (\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, p, e, g, G_z, G_r, H_z, H_r, (G_i, H_i)_{i=1}^n, A, B)$
  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．

- **AHOSign**
  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 $(M_1, \cdots, M_n) \in \mathbb{G}^n$ に署名するために，ランダムに選ばれた $\mu, \rho_a, \rho_b, \omega_a, \omega_b \in \mathbb{Z}_p$ を設定し, 以下のように署名を作成する．

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

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

- **AHOVerify**
  署名 $\sigma = (\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6, \theta_7)$ が以下の式を満たしていれば正当な署名として受理する．

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

上記の方式では，メッセージは $\mathbb{G}_2$ の要素となるが，各パラメータを $\mathbb{G}_1$ 要素，$\mathbb{G}_2$ 要素で入れ替えることにより，$\mathbb{G}_1$ 要素にも署名ができる．この入れ替えた方式も利用する．

2.6 ベクトルコミットメント

ベクトルコミットメント [8] はベクトルの要素を圧縮してコミットメント化する方式であり，圧縮されたベクトルの何番目にその要素があるかを証明することができる。
まず，圧縮に用いる公開鍵

$$ck = (g, g_1, ..., g_l, g_{l+2}, ..., g_{2l}, \tilde{g_1}, \tilde{g_2}, \cdots, \tilde{g_l}, \tilde{g_{l+2}}, \cdots, \tilde{g_{2l}})$$

($\alpha \xleftarrow{R} \mathbb{Z}_p$ に対して，$g_i = g^{(\alpha^i)}, \tilde{g_i} = \tilde{g}^{(\alpha^i)}$) を生成する．そして，ベクトル $\vec{m} = (m_1, ..., m_l)$ を圧縮するために，

$$C = \Pi_{k=1}^{l} \tilde{g}_{l+1-k}^{m_k}$$

を計算する。この C が圧縮された値となる。
圧縮前のベクトルの i 番目の要素が m_i であることを証明するためには，まず，証拠情報 $W_i = \Pi_{k=1, k\neq i}^{l} \tilde{g}_{l+1-k+i}^{m_k}$ を計算し，$e(g_i, C) = e(g, W_i) e(g_1, \tilde{g}_l)^{m_i}$ が成り立つことを証明すれば良い．この方式は，n-DHE 仮定の基で安全である．

3. 従来方式とその問題点

従来方式 [3] では木構造による CS(Complete Subtree) 法を用いて効率的な失効を実現している. 以下ではその説明を行う.

3.1 従来方式のセットアップ

事前に GM(Group Manager) は高さを a として葉の数が 2^a となるような二分木を生成する. そして, 根を 0 として順にノード番号を図 1 のように割り振り, 各葉にユーザを

割り当てる. 具体的に図 1 の場合では 7 から 14 がユーザに該当する. また GM は, 新たにユーザを加える際に各ユーザの葉から根へのパス上のノード ID $(u_0, u_1, \cdots, u_l)$ に対して, それぞれ証明書 A_i を作成して，ユーザに発行する.

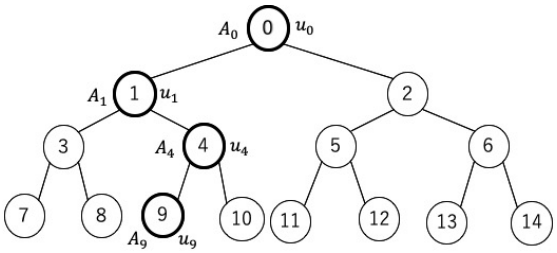

図 1　ユーザ数 8 の場合の二分木

3.2　従来方式の失効

失効が発生すると, GM は CS 法を用いて, 失効されるユーザだけが所持せず, それ以外の正規のユーザが所持しているノード ID の集合を探し, その各ノード ID に対して証明書 B_j を生成する. こうすることにより, ユーザは自身の所持している証明書 A_i と GM が生成した証明書 B_j を比較し, 同じノード u_i で証明書が生成されていることを証明することで, ユーザが失効されていないことを保証できる. この方式の効率的である点は, CS 法により正規のユーザのみが属するできるだけ大きな部分木を取ることで失効リストを減らすことができるという点である.

図 2 の場合, ユーザ 9 を失効すると, GM が CS 法で選んだノード ID は (u_2, u_3, u_{10}) となり, 生成される証明書は (B_2, B_3, B_{10}) となる. そして, 失効リストは (B_2, B_3, B_{10}) から成る.

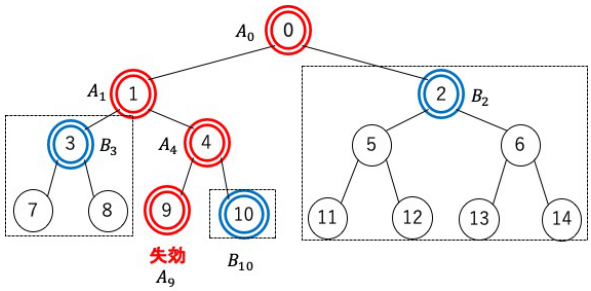

図 2　ユーザ数 8 の場合の失効リストの生成

3.3　従来方式の署名

署名者は署名の際, 失効リストを GM から取得して, 署名者の葉から根までのパスから作られた証明書と失効リスト中の証明書とを比較し, 同じノード ID から生成されている A_i と B_j を取り出す. 例として, 署名者が持つ証明書 A_i がノード u_i から生成され, 失効リスト中の証明書 B_j がノード $u_j^{'}$ から生成されているとする. そして署名者は $u_i = u_j^{'}$ を証明することで, 署名者は失効されていないユーザであることを示すことができる. この関係をゼロ知識証明を利用して証明するため, 署名者のユーザ情報の匿名性を保つことができる.

3.4　従来方式の問題点

従来方式では, r を失効ユーザ数, N を全ユーザ数として失効リストのサイズは $\mathcal{O}(r \log \frac{N}{r})$ となり, 失効ユーザ数に大きく依存してしまう. そして失効ユーザ数が増えると失効リストも増大し, そのデータサイズが問題となる.

4.　提案方式

4.1　提案方式の概要

本研究では失効リストを削減するために, ベクトルコミットメントを用いて従来方式の拡張を行う. 以下に提案方式について示す.

まず, GM は CS 法でのノード ID の集合 $\{u_0^{'}, u_1^{'}, \cdots, u_{num}^{'}\}$ をベクトル $\vec{u}$ として設定する.

$$\vec{u} = (u_0^{'}, u_1^{'}, \cdots, u_{num}^{'})$$

そして, GM は分割パラメータ T を設定し, $\vec{u}$ の要素を T ごとに分割する.

$$\vec{u}_1 = (u_0^{'}, u_1^{'}, \cdots, u_T^{'})$$
$$\vec{u}_2 = (u_{T+1}^{'}, u_{T+2}^{'}, \cdots, u_{2T}^{'})$$
$$\vdots$$
$$\vec{u}_{num/T} = (u_{(num/T-1)T+1}^{'}, u_{(num/T-1)T+2}^{'}, \cdots, u_{num}^{'})$$

分割されたベクトル $\vec{u}_1, \vec{u}_2, \cdots, \vec{u}_{num/T}$ それぞれに対して, ベクトルコミットメントによって以下のように圧縮を行う.

$$C_{\vec{u}_1} = \prod_{j=1}^{T} \tilde{g}_{n+1-j}^{u_j^{'}}, C_{\vec{u}_2} = \prod_{j=1}^{T} \tilde{g}_{n+1-j}^{u_{T+j}^{'}}$$
$$\vdots$$
$$C_{\vec{u}_{num}} = \prod_{j=1}^{T} \tilde{g}_{n+1-j}^{u_{(num-1)T+j}^{'}}$$

ベクトルコミットメントにより圧縮された $C_{\vec{u}_k}$ を AHO 署

名で署名したものを提案方式の失効リストとする. これにより従来方式と比べて, 失効リストのサイズは $\mathcal{O}(r\log\frac{N}{r})$ から $\mathcal{O}(\frac{r\log\frac{N}{r}}{T})$ に削減されている.

グループ署名の際には, ベクトルコミットメントのベクトルの要素を証明できることを利用する. 署名者が所持している証明書 A のノード ID $(u_0, u_1, \cdots, u_\ell)$ と失効リスト中の署名者が含まれている部分木の頂点のノード ID $u_j^{'}$ を含むベクトルコミットメントを抽出する. そして証明書 A 中のノード ID u_i と失効リスト中の $u_j^{'}$ が $u_i = u_j^{'}$ であるということを証明するとともに, $u_j^{'}$ が抽出したベクトルコミットメントに含まれていることを証明することにより失効されていないことを示すことができる.

4.2 提案方式のアルゴリズム

- **Setup**

(1) 従来方式 [3] と同様に, 双線形群のパラメータ $(p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, g, h)$ を選び, $f_1, f_2, f_3, h_0, h_1, h_2 \leftarrow \mathbb{G}_1/\{1\}$ をランダムに選ぶ. $\gamma_0 \leftarrow \mathbb{Z}_p$ をランダムに選び, BBS+署名の鍵を $(sk_0, vk_0) = (\gamma_0, h^{\gamma_0})$ とする. 次に $\xi_1, \xi_2, \xi_3 \leftarrow \mathbb{Z}_p$ をランダムに選び, $g_1^{'} = f_1^{\xi_1} f_3^{\xi_3}, g_2^{'} = f_2^{\xi_2} f_3^{\xi_3}$ を計算する. ハッシュ関数 $H : \{0,1\}^* \rightarrow \mathbb{Z}_p$ を選ぶ. そして, $sk_{OA} = (\xi_1, \xi_2, \xi_3)$, $St = (St_{users}, St_{trans}) = (0, \emptyset)$ とする.

(2) $\delta_0 = 1$, $\delta_1 = 2$ とする. δ_0 個, δ_1 個のメッセージにそれぞれ署名するため, $d = 0, 1$ に対して, AHO 署名用の 2 つの鍵ペア $(pk_{AHO}^{(d)}, sk_{AHO}^{(d)})$ を生成する. それぞれの鍵ペアは以下のようになる.

$$pk_{AHO}^{(d)} = (G_r^{(d)}, H_r^{(d)}, G_z^{(d)} = G_r^{\gamma_z^{(d)}}, H_z^{(d)} = H_r^{\delta_z^{(d)}}, \{G_i^{(d)} = G_r^{\gamma_i^{(d)}}, H_i^{(d)} = H_r^{\delta_i^{(d)}}\}_{i=1}^{n_d}),$$
$$sk_{AHO}^{(d)} = (\alpha_a^{(d)}, \alpha_b^{(d)}, \gamma_z^{(d)}, \delta_z^{(d)}, (\gamma_i^{(d)}, \delta_i^{(d)})_{i=1}^{n_d}) \quad (d = 0, 1)$$

(3) $n(= T)$ 次元のベクトルコミットメントのための公開鍵 pk_{vc} を $\mathbb{G}_1, \mathbb{G}_2$ からそれぞれ生成する.

$$pk_{vc} = (g_1, g_2, \cdots, g_n, g_{n+2}, \cdots, g_{2n}, \tilde{g}_1, \tilde{g}_2, \cdots, \tilde{g}_n, \tilde{g}_{n+2}, \cdots, \tilde{g}_{2n})$$

(4) $sk_{AHO}^{(0)}$ を用いて, メッセージ $\tilde{g}_j$ の AHO 署名 $\sigma_j = (\theta_{\tilde{j},1}, \cdots, \theta_{\tilde{j},7})$ を生成する. ここで, $1 \le j \le T$ とする.

(5) コミットメント用の $S \in \mathbb{G}_1, \tilde{S} \in \mathbb{G}_2$ をランダムに選ぶ。

(6) グループの公開鍵は,

$$gpk = (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, g, f_1, f_2, f_3, h_0, h_1, h_2, h, vk_0, H, pk_{AHO}^{(0)}, pk_{AHO}^{(1)}, pk_{vc}, \{\sigma_j\}_{j=1}^{T}, S, \tilde{S})$$

となる.

- **Join**

(1) ユーザ i は $x \leftarrow \mathbb{Z}_p$ をランダムに選び, メッセージ $X = h_2^x$ に対する通常のディジタル署名 sig_i を計算し, GM に (X, sig_i) を送る.

(2) GM はユーザ i を二分木の葉 u_ℓ に割り当てる. ここで, $u_0, u_1, \cdots, u_\ell$ はルートノードからその葉ノード u_ℓ へのパス上のノードとする. $j = 0, \cdots, \ell$ として, GM は $\eta_j, \zeta_j \leftarrow \mathbb{Z}_p$ をランダムに選び,

$$A_j = (g h_0^{\zeta_j} h_1^{u_j} X)^{\frac{1}{\gamma_0 + \eta_j}}$$

を計算し, ユーザ i に $\{\theta_j = (A_j, \eta_j, \zeta_j)\}_{j=0}^{\ell}$ と $\langle v_i \rangle := (u_0, u_1, \cdots, u_\ell)$ を送る.

(3) ユーザはメンバーシップ証明書 $cert_i = (\langle v_i \rangle, \{A_j\}_{j=0}^{\ell}, X)$ と秘密鍵 $sec_i = x$ を取得する.

(4) 最後に GM は i と $transcript_i = (X, \{A_j\}_{j=0}^{\ell}, sig_i)$ を St_{trans} に追加する.

- **Revoke**

(1) 現在の木に対して, CS 法を適用して得られる部分木のルートノードを $\{u_0^{'}, u_1^{'}, \cdots, u_{num}^{'}\}$ とする. ここで $(num \le r\log\frac{N}{r})$ である.

(2) ベクトル $\vec{u} = (u_0^{'}, u_1^{'}, \cdots, u_{num}^{'})$ とし, 分割パラメータ T を決め,$\vec{u}$ を T ごとに分割する. このとき $\Omega = num/T$ 個のベクトルに分割されることになる.

$$\vec{u_1} = (u_0^{'}, u_1^{'}, \cdots, u_T^{'})$$
$$\vec{u_2} = (u_T^{'}, u_{T+1}^{'}, \cdots, u_{2T}^{'})$$
$$\vec{u_\Omega} = (u_{(\Omega-1)T+1}^{'}, u_{(\Omega-1)T+2}^{'}, \cdots, u_{num}^{'})$$

となる.

(3) ベクトルコミットメントを用いて, $(1 \le k \le \Omega)$

に対して

$$C_{\vec{u}_k} = \prod_{j=1}^{T} \tilde{g}_{n+1-j}^{u'_{(k-1)T+j}}$$

を計算する．

(4) AHO 署名の秘密鍵 $sk_{AHO}^{(1)}$ を用いて, ステップ 3 で圧縮した $C_{u\vec{k}}$ を時刻 t と共に署名して，AHO 署名 $\sigma_{RL_k} = (\Theta_{k,1}, \cdots, \Theta_{k,7})$ を生成する．ここで，$(1 \leq k \leq \Omega)$ である．以下の失効リストを出力する．

$$RL_t = (t, R_t, \vec{u}, \{C_{u\vec{k}}, \sigma_{RL_k} = (\Theta_{k,1}, \cdots, \Theta_{k,7})\}_{k=1}^{\Omega})$$

- **Sign**

(1) 現在の時刻 t の RL_t を用いて，署名者の葉ノード u_i を含んでいるベクトルコミットメント $C_{u\vec{k}}$ を探す．なお，$C_{u\vec{k}}$ 中の $\tilde{j}$ 番目の要素は u_i に含まれてるとする．すなわち $i = (k-1)T + \tilde{j}$ すると，$u_{k,\tilde{j}} = u_i$ として定義する．$\tilde{j}$ 番目の要素を証明するために，以下の証拠情報 $W_{\vec{u}_k}$ を計算する．

$$W_{\vec{u}_k} = \prod_{j=1, j\neq\tilde{j}}^{T} \tilde{g}_{n+1-j+\tilde{j}}^{u_{k,\tilde{j}}}$$

そして，$g_{\tilde{j}}, C_{\vec{u}_k}, W_{u\vec{k}}$ のコミットメント $C_{g_{\tilde{j}}}, C_{C_{u\vec{k}}}, C_{W_{u\vec{k}}}$ を乱数 $r_{g_{\tilde{j}}} \in \mathbb{Z}_p$ に対して $C_{g_{\tilde{j}}} = g_{\tilde{j}} S^{r_{g_{\tilde{j}}}}$（他も同様）として計算し，以下のベクトルコミットメントの検証式を示す SPK を計算して正しさを証明する．この SPK を $\pi_{C_{\vec{u}_k}}$ とする．

$$\begin{aligned} &e(C_{g_{\tilde{j}}}, C_{C_{u\vec{k}}}) \cdot e(g, C_{W_{u\vec{k}}})^{-1} = e(g_1, g_n)^{u_i} \\ &\cdot e(C_{g_{\tilde{j}}}, \tilde{S}^{-1})^{-r_{C_{u\vec{k}}}} \cdot e(S, C_{C_{u\vec{k}}})^{r_{g_{\tilde{j}}}} \cdot e(g, \tilde{S}^{-1})^{r_{W_{u\vec{k}}}} \\ &\cdot e(S, \tilde{S}^{-1})^{r_{g_{\tilde{j}}} \cdot r_{C_{u\vec{k}}}} \end{aligned}$$

(2) 以下のように $\alpha = r_{g_{\tilde{j}}} \cdot r_{C_{u\vec{k}}}$ を証明する．$r_{C_{u\vec{k}}}, \alpha$ に対するコミットメント $C_{r_{C_{u\vec{k}}}}, C_\alpha$ を $\rho_1, \rho_2 \in \mathbb{Z}_p$ をランダムに選び、以下のように生成する．

$$\begin{aligned} C_{r_{C_{u\vec{k}}}} &= R^{r_{C_{u\vec{k}}}} S^{\rho_1} \\ C_\alpha &= R^\alpha S^{\rho_2} \end{aligned}$$

そして上記の 3 式と以下の知識を証明する．

$$C_\alpha = C_{r_{C_{u\vec{k}}}}^{r_{g_{\tilde{j}}}} S^{\rho'}$$

(3) 従来方式 [3] と同様に署名者 (ユーザ) が持つメンバーシップ証明書の A_j に含まれている u_j と $C_{\vec{u}_k}$ の $\tilde{j}$ 番目に含まれている u_i に対して $u_j = u_i$ を証明することにより，ユーザは失効されていないことを証明する．まず初めに $\alpha, \beta \leftarrow \mathbb{Z}_p$ をランダムに選ぶ．そして，$\psi_1 = f_1^\alpha, \psi_2 = f_2^\beta, \psi_3 = f_3^{\alpha+\beta}, \psi_4 = (g_1^\alpha g_2^\beta A_j)$ を計算する．次に署名者は (A_j, x) を保持していることと，u_j は A_j に対応することを示す以下の式に対する SPK を計算する．この SPK を π_{A_j} とする．

$$\begin{aligned} \psi_1 = f_1^{r_\alpha}, \psi_2 = f_2^{r_\beta}, \psi_3 = f_3^{r_\alpha + r_\beta} \\ e(\psi_4 \cdot g_1^{'-r_\alpha} g_2^{'-r_\beta}, h^{r_\eta} v k_0) = e(g h_0^{r_\zeta} h_1^{r_{u_j}} h_2^{r_x}, h) \\ \psi_1^{r_\eta} f_1^{-r_{\alpha\eta}} = 1, \psi_2^{r_\eta} f_2^{-r_{\beta\eta}} = 1 \end{aligned}$$

(4) $g_{\tilde{j}}$ の正しさを保証するため，AHO 署名 $\sigma_{g_{\tilde{j}}}$ を $(\theta'_1, \cdots, \theta'_7)$ に再ランダム化し，$\{\theta'_j\}_{j=\{1,2,5\}}$ のコミットメント $\{C_{\theta'_j}\}_{j=\{1,2,5\}}$ を乱数 $r_{\theta'_j} \in \mathbb{Z}_p$ に対して $C_{\theta'_j} = \theta'_j S^{r_{\theta'_j}}$ として計算し以下の検証式を示す SPK を計算する．この SPK を $\pi_{g_{\tilde{j}}}$ とする．

$$\begin{aligned} &A \cdot e(C_{\theta'_1}, G_z)^{-1} \cdot e(C_{\theta'_2}, G_r)^{-1} \\ &\cdot e(\theta'_{\tilde{j},4}, \theta'_{\tilde{j},3})^{-1} \cdot e(C_{C_{g_{\tilde{j}}}}, G_1)^{-1} \\ &= e(S^{-1}, G_z)^{r_{\theta'_1}} \cdot e(S^{-1}, G_r)^{r_{\theta'_2}} \cdot e(S^{-1}, G_1)^{r_{C_{g_{\tilde{j}}}}}, \\ &B \cdot e(C_{\theta'_1}, H_z)^{-1} \cdot e(C_{\theta'_5}, H_r)^{-1} \\ &\cdot e(\theta'_{\tilde{j},7}, \theta'_{\tilde{j},6})^{-1} \cdot e(C_{C_{g_{\tilde{j}}}}, H_1)^{-1} \\ &= e(S^{-1}, H_z)^{r_{\theta'_1}} \cdot e(S^{-1}, H_r)^{r_{\theta'_5}} \cdot e(S^{-1}, H_1)^{r_{C_{g_{\tilde{j}}}}} \end{aligned}$$

(5) $C_{\vec{u}_k}$ が正しい RL_t から導出されたことを証明するために，AHO 署名 σ_{RL_t} を $(\Theta'_1, \cdots, \Theta'_7)$ に再ランダム化し，$C_{\vec{u}_k}$ と $\{\Theta'_j\}_{j\in\{1,2,5\}}$ に対するコミットメント $\{C_{\Theta'_j}\}_{j\in\{1,2,5\}}$ を乱数 $r_{\Theta'_j} \in \mathbb{Z}_p$ に対して $C_{\Theta'_j} = \Theta'_j S^{r_{\Theta'_j}}$ として計算し，以下の検証式を示す SPK を計算する．この SPK を π_{RL_t} とする．

$$\begin{aligned} &A \cdot e(G_z, C_{\Theta'_1})^{-1} \cdot e(G_r, C_{\Theta'_2})^{-1} \cdot e(\Theta'_3, \Theta'_4)^{-1} \\ &\cdot e(G_1, C_{C_{u\vec{k}}})^{-1} \cdot e(G_2, g^t)^{-1} \\ &= e(G_z, \tilde{S}^{-1})^{r_{\Theta'_1}} \cdot e(G_r, \tilde{S}^{-1})^{r_{\Theta'_2}} \cdot e(G_1, \tilde{S}^{-1})^{r_{C_{u\vec{k}}}} \\ &B \cdot e(H_z, C_{\Theta'_1})^{-1} \cdot e(H_r, C_{\Theta'_5})^{-1} \cdot e(\Theta'_6, \Theta'_7)^{-1} \\ &\cdot e(H_1, C_{C_{u\vec{k}}})^{-1} \cdot e(H_2, g^t)^{-1} \\ &= e(H_z, \tilde{S}^{-1})^{r_{\Theta'_1}} \cdot e(H_r, \tilde{S}^{-1})^{r_{\Theta'_5}} \cdot e(H_1, \tilde{S}^{-1})^{r_{C_{u\vec{k}}}} \end{aligned}$$

(6) 署名 $\sigma = (\psi_1, \psi_2, \psi_3, \psi_4, C_{C_{u_k^-}}, C_{W_{u_k^-}}, C_{g_{\tilde{j}}}, C_{r_{C_{u_k^-}}}, C_\alpha, C_{\theta_1'}, C_{\theta_2'}, C_{\theta_5'}, \theta_3', \theta_4', \theta_6', \theta_7', C_{\Theta_1'}, C_{\Theta_2'}, C_{\Theta_5'}, \Theta_3', \Theta_4', \Theta_6', \Theta_7', \pi_{C_{\vec{u}_k}}, \pi_{A_j}, \pi_{g_{\tilde{j}}}, \pi_{RL_t})$ を出力する．

- **Verify**
  検証者は署名の中の各 SPK を検証する．全ての SPK が正しいならば，1 を出力する．

- **Open**
  $A' = (\psi_4/\psi_1^{\xi_1}\psi_2^{\xi_2}\psi_3^{\xi_3})$ を計算する．もし St_{trans} において $A' = A_j$ となる $\langle i, taranscript_i \rangle = (X, \{A_j\}_{j=0}^{\ell}, sig_i)$ が存在するならば，sig_i を検証し，sig_i が有効な署名ならば i を出力する．

5. 安全性

4 章で提案した方式の安全性は以下のような観点から達成できている．

Anonymity, Unlinkability： グループ署名中の各情報は，従来方式の暗号文 $(\psi_1, \psi_2, \psi_3, \psi_4)$ に加えて，コミットメント，ランダマイズした署名，SPK のみであり情報が漏れないため，ユーザを特定することはできない．また，同様に同じユーザの署名かどうかも分からない．

Traceability： グループ署名では GM のみ作成可能な署名 A_j を SPK で証明しており，グループ外のユーザが作成することはできない．よって **Open** の処理内で各ユーザが作成した署名の $\psi_1, \psi_2, \psi_3, \psi_4$ と GM のみが所持している $sk_{OA} = (\xi_1, \xi_2, \xi_3)$ を用いて，$A' = (\psi_4/\psi_1^{\xi_1}\psi_2^{\xi_2}\psi_3^{\xi_3})$ を計算した結果，ユーザが所持している証明書 $A_j = A'$ となるユーザを探しているため，その署名者 i を正しく特定できる．

また，**Sign** の (3) で従来方式と同様に A_j で証明された u_j と失効リスト RL 中の u_i が等しいことの証明をしており，u_i が正しいなら非失効が証明される。一方，**Sign** の (1) では u_i があるベクトルコミットメント $C_{\vec{u}_k}$ に入っていることを検証しており，検証式中の $g_{\tilde{j}}$ の正しさも (4) で AHO 署名の SPK により保証されている．またそのベクトルコミットメントが AHO 署名で時刻 t において署名されていることも (5) で検証しているので、u_i は時刻 t での CS 法の部分木に入っていることになる．

6. 実装結果

6.1 実験環境

提案方式の認証時間における有効性を示すために表 1 の環境で提案方式の実装を行い，署名生成，検証，証拠情報 W の計算時間の測定を行った．また従来方式 [3] と提案方式の失効リストのサイズおよび，署名長も比較している．128bit 安全性のため，各 $\mathbb{G}_1, \mathbb{G}_2$ 要素，$\mathbb{Z}_p$ 要素は 512bit としている．

表 1 実装環境

OS	Ubuntu 18.04.3 LTS
CPU	Intel Core i5-9600 (3.70GHz)
メモリ	7.7GB
ライブラリ	GMP6.1.2 (Arithmetic library)，ELiPS (Pairing library)[9]

6.2 実験結果

6.2.1 失効リストのサイズの比較

まず，失効リストサイズを比較する．分割パラメータは $T = 100, 500, 1,000$ として，全体のユーザ数 N，失効ユーザ数 R を $\frac{R}{N} = 0.1$ に固定して変化させた時の計測結果を表 2 に示す．従来方式では失効ユーザ数 R に依存していて増大しており，$R = 100,000$ では 40MB 以上となっている．提案方式も失効ユーザ数に依存しているものの，$R = 100,000$ でも $T = 1,000$ において 1MB 程度となっており，失効リストサイズが削減できている．さらに分割パラメータを $T = 1,000$ に増やすことにより $R = 100,000$ の場合で 0.1MB 程度に削減できる．

6.2.2 処理時間の比較

分割パラメータ T を変化させた時の提案方式における **Sign**, **Verify**, **Sign** および **Verify**，ベクトルコミットメントの証拠情報である W の計算時間についての処理時間を図 3 に示す．**Verify** の処理時間は一定であるが，**Sign** の計算時間は分割パラメータ T の増加によって増加している．これは，**Sign** の中で行われている証拠情報 W の計算が T に依存しているためである．**Sign** の処理時間は増大するものの $T = 1000$ の場合でも 200ms 程度であり，処理時間は十分実用的であるといえる．

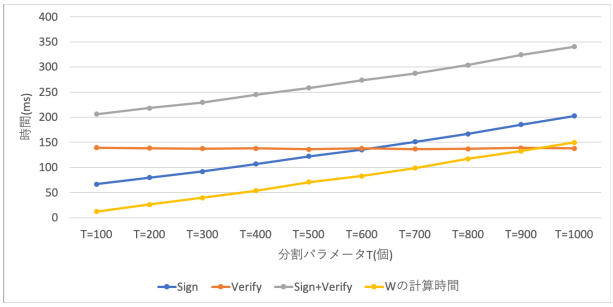

図 3 処理時間の比較

表 2　失効リストのサイズ

	従来方式 [3]	提案方式		
		T=100	T=500	T=1,000
N=10,000(R=1,000)	442KB	12KB	2KB	1KB
N=100,000(R=10,000)	4,421KB	118KB	24KB	12KB
N=1,000,000(R=100,000)	44,210KB	1179KB	236KB	118KB

表 3　署名長・公開鍵のサイズ比較

	Sign 中のコミットメント数	Sign 中の SPK のパラメータ数	署名長	公開鍵サイズ
従来方式 [3]	5	13	9,216bit	1.28KB
提案方式	30	23	27,136bit	321KB(T=1,000)

6.2.3　署名長の比較

署名長，公開鍵のサイズの比較を表 3 に示す. 従来方式と比べて提案方式は, **Sign** 中のコミットメント数, SPK 中の要素のどちらも上回っており署名長も長い. しかしこれはベクトルコミットメントの採用により，その検証や AHO 署名の検証が追加されたためである. したがって失効リストのサイズと署名長はトレードオフの関係にある. 公開鍵のサイズは分割パラメータ T に依存するため, 従来方式よりも大きくなるが，$T = 1,000$ の場合では 321KB 程度と十分実用的である.

7.　まとめ

本研究では, 効率的に失効可能な従来方式 [3] の失効リストを削減する方式を提案した. 提案方式では, 分割パラメータ T を設定し, ノード ID の集合 $\{u_0^{'}, u_1^{'}, \cdots, u_{num}^{'}\}$ を T 個に分割して, ベクトルコミットメントを用いて圧縮することにより失効リストの削減を行った. また実装を行った結果, **Sign**, **Verify** の時間は十分実用性があるという結果が得られた. しかし, 失効リストのサイズを削減するということのトレードオフとして **Sign**/**Verify** の時間や処理時間や署名長が長くなってしまう. 今後の課題としては失効リストを削減したまま署名長を小さくすることが挙げられる.

参考文献

[1] D. Chaum and E. van Heyst, "Group Signatures," EUROCRYPT '91, LNCS547, pp.257-265, Springer-Verlag, 1991.

[2] B. Libert, T. Peters and M. Yung, "Scalable group signatures with revocation," EUROCRYPT 2012, LNCS7323, pp.609-627, Springer-Verlag, 2012.

[3] K. Ohara, K. Emura, G. Hanaoka, A. Ishida, K. Ohta, and Y. Sakai "Shortening the Libert-Peters-Yung Revocable Group Signature Scheme by Using the Ramdom Oracle Methodology ", IEICE Trans. Fundamentals, Vol.E102-A, No.9, pp.1101-1117, 2019.

[4] S. Sadiah and T. Nakanishi,"Revocable group signatures with compact revocation list using vector commitments," IEICE Trans. Fundamentals, Vol.100-A, No.8, pp.1672-1682, 2017.

[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] D. Boneh, X. Boyen, and H. Shacham, "Short group signatures,"CRYPTO 2004, LNCS3152, pp.41-55, Springer-Verlag, 2004.

[7] M. H. Au, W. Susilo, and S. S. M. Chow, "Constant-size dynamic k-times anonymous authentication," IEEE Systems Journal, Vol.7, No.2, pp.249-261, 2013.

[8] B.Libert and M.Yung, "Concise mercurial vector commitments and independent zero-knowledge sets with short proofs," TCC 2010, LNCS 5978, pp.499-517, Springer-Verlag, 2010.

[9] M. Akane, Y. Nogami, Y. Morikawa, "Fast Ate Pairing Computation of Embedding Degree 12 Using Subfield-Twisted Elliptic Curve", IEICE Trans. Fundamentals, Vol.E92-A, No.2, pp.508-516, 2009.

[10] D. Boneh and H. Shacham, "Group Signatures with Verifier-Local Revocation", ACM Conference on Computer and Communications Security(CCS2004), pp.168-177, 2004.