分散ブラックリスト型匿名認証システムにおける認証データサイズの削減

入星　敦輝[1,a)]　中西　透[1,b)]

概要：プライバシ保護のために，ID を秘匿しながらユーザの正当性を証明できる匿名認証システムとして，ブラックリスト型匿名認証が提案されている．しかし，ユーザ認証時に用いるチケットが，資格発行者が発行した証明書に基づいており，資格発行者の信頼が必要である．ブロックチェーンへの応用ではそのようなエンティティを仮定できないため，分散ブラックリスト型匿名認証が提案されている．その従来方式では，ユーザが認証する際，自身の公開鍵を含むユーザの公開鍵集合を用いる．公開鍵はブロックチェーン上にあり，自身の公開鍵がユーザ集合に含まれることを検証することで，資格発行者なしで自身が正しいユーザであることを証明できる．しかし従来方式では，ユーザ集合すべての公開鍵と照合して OR 証明するため，認証データサイズがユーザ数に比例してしまい，効率が悪い．

そこで本研究では，従来のブラックリスト型匿名認証に効率的なリング署名を組み合わせることで，分散ブラックリスト型匿名認証の認証データサイズをユーザ数の log サイズまで削減した方式を提案する．

キーワード：匿名認証，リング署名，ブロックチェーン，プライバシ保護

Reducing Authentication Data Size in Decentralized Blacklistable Anonymous Credential System

Atsuki Iriboshi[1,a)]　Toru Nakanishi[1,b)]

Abstract: For privacy protection, blacklistable anonymous credential systems have been proposed, where a user is authenticated without ID. However, the trust for the credential issuer is needed. Since such an entity cannot be assumed in public blockchain, a decentralized system has been proposed. In this previous system, a user is authenticated using a set of public-keys of users. Since the public-keys are on the blockchain, a user can prove that he is valid without a credential issuer by verifying that his public-key is included in the public-key set. However, this previous system is inefficient, because the size of the authentication data is proportional to the number of users, due to an OR proof with the public-keys.

In this paper, we propose a system where the size of the authentication data is reduced to the log-order size, by combining a conventional blacklistable anonymous credential system with an efficient ring signature scheme.

Keywords: Anonymous credential systems, Ring signatures, Blockchain, Privacy protection

1. はじめに

現在，ユーザとサービス提供者 (以降 SP) 間の通信において，ID ベース認証が広く利用されている．しかし， SP はユーザ ID からユーザ個人のサービス利用履歴を追跡で

[1] 広島大学
Hiroshima University
[a)] m216877@hiroshima-u.ac.jp
[b)] t-nakanishi@hiroshima-u.ac.jp

きてしまうため，プライバシの観点から問題がある．

そこで， ID を秘匿して自身が正規ユーザであることのみを証明できる匿名認証システムとして，ブラックリスト型匿名認証 [1] が提案されている．[1] では，ユーザは自身の秘密鍵からチケットを生成し，それを用いて SP に対して認証を行う．また， SP は過去に不正を行ったユーザのチケットを記録しているブラックリストを所持しており，匿名不正者排除のために，資格発行者を利用することなくブラックリストを用いて不正ユーザを排除することができる．しかしこの方式では，認証に用いるチケットの正当性を保証するために，資格発行者が発行した証明書に基づいてチケットが生成されるため，資格発行者の信頼が必要である．ブロックチェーンへの応用では，そのようなエンティティは仮定できないため，この問題を解決するために，分散ブラックリスト型匿名認証 [2] が提案されている．[2] では，ユーザが認証する際，自身の公開鍵に加えて他のユーザの公開鍵集合を用いて認証を行う．公開鍵はブロックチェーン上にあり，改ざんが難しいため，自身の公開鍵がユーザ集合に含まれることを検証することで，資格発行者が証明書を発行しなくても自身が正しいユーザであることを証明できる．しかしこの方式のインスタンス化において，RSA をベースとした場合， RSA アキュムレータを利用することにより効率化している．しかし，短データサイズを実現する楕円曲線暗号ベースでは，単純な OR 型証明しか利用できず，認証に用いる公開鍵集合の中に自身の公開鍵が含まれているかどうかを検証する際，ユーザ集合すべての公開鍵と照合してゼロ知識証明しなければならないため，認証データサイズがユーザ人数に比例してしまい，効率が悪いという問題が存在する．

そこで本研究では， [1] の方式に効率的なリング署名 [3] を組み合わせる．リング署名 [4] とは，自身の秘密鍵を用いて署名を生成する際に，自身の公開鍵に加え，他のユーザの公開鍵集合を用いて，その中に自身の公開鍵が含まれることを証明することにより，匿名性を保証しながら署名できる方式である．オリジナルのリング署名方式 [4] では，全ユーザ数に比例した署名長となるが，2015 年に提案された [3] の方式では，全ユーザ数に対して log サイズの署名長となっている．そこで，[3] のリング署名で用いられている公開鍵集合に含まれることを示すゼロ知識証明をブラックリスト型匿名認証 [1] と組み合わせることで，認証データサイズを log サイズまで削減した方式を提案する．さらに， PC 上で認証のデータサイズと処理時間を計測し，その有効性を検証する．

2. 数学的準備

2.1 安全性仮定

提案方式は，[2]，[3] と同様に，以下の仮定に基づく．

- **離散対数 (DL) 仮定：**$\mathbb{G}$ を位数が大きな素数 q の巡回群とし，その生成元を g とする．$y = g^x$ について，g，y，q が与えられたとき，x を無視できない確率で求める確率的多項式時間アルゴリズムが存在しない．
- **Decisional Diffie-Hellman(DDH) 仮定：**$\mathbb{G}$ を位数が大きな素数 q の巡回群とし，その生成元を g とする．a，b，$c \in \mathbb{Z}_q$ をランダムに選んだとき，(g, g^a, g^b, g^{ab}) と (g, g^a, g^b, g^c) は多項式時間において識別不可能である．

2.2 ElGamal 暗号

ElGamal 暗号 [5] は， DDH 仮定の下で強秘匿性を持つ公開鍵暗号であり，以下のアルゴリズムから成る．

- **鍵生成アルゴリズム：**秘密鍵 $sk \in \mathbb{Z}_q$ をランダムに選び，公開鍵を $pk = g^{sk}$ とする．
- **暗号化アルゴリズム：**平文 $m \in \langle g \rangle$に対し， $r \in \mathbb{Z}_q$ をランダムに選び，暗号文 (c_1, c_2) を以下のように計算する．

$$\mathrm{Enc}_{pk} = (c_1, c_2) = (g^r, m \cdot pk^r)$$

- **復号アルゴリズム：**暗号文 (c_1, c_2) および秘密鍵 x から，平文 m を以下のように計算する．

$$\begin{aligned}\mathrm{Dec}_{sk}(c_1, c_2) &= c_2 / c_1^{sk} = m \cdot pk^r / g^{r \cdot sk} \\ &= m g^{r \cdot sk} / g^{r \cdot sk} = m\end{aligned}$$

2.3 コミットメント方式

複数のメッセージを圧縮してコミットメント化する方式であり，2 つの多項式時間アルゴリズム (CGen，Com_{ck}) で構成される．CGen はコミットメント鍵 ck を出力し，Com_{ck} は，メッセージ m と乱数 r を入力として，コミットメント c を出力する確率的アルゴリズムである．
$r \leftarrow \mathbb{Z}_q$，$h \leftarrow \mathbb{G}$，$m \in \mathbb{Z}_q$ で $c = g^r h^m$ となるコミットメントを Pedersen コミットメント [6] という．Pedersen コミットメントを拡張した複数メッセージ $m_1, ..., m_n \in \mathbb{Z}_q$ に対するコミットメントのアルゴリズムを以下に示す．

- **CGen：**$h_1, ..., h_n \leftarrow \mathbb{G}$ を選び，$ck = (h_1, ..., h_n)$ を出力する．
- $\mathbf{Com_{ck}(m_1, ..., m_n)}$：$r \leftarrow \mathbb{Z}_q$ を選び，$c := g^r \prod_{i=1}^{n} h_i^{m_i}$ を出力する．

2.4 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 $\mathcal{P}$ と検証者 $\mathcal{V}$ の対話型プロトコルで，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．

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

2.4.1 リプレゼンテーションの SPK

$y = g^x$ となる離散対数の関係は，複数の底 g_1，g_2 を導入することで，秘密情報 x_1，x_2 に対し，$y = g_1^{x_1} g_2^{x_2}$ のように拡張できる．以下に $SPK\{(x_1, x_2) : y = g_1^{x_1} g_2^{x_2}\}(m)$ を示す．ここで H はハッシュ関数である．

秘密鍵： x_1，$x_2 \in \mathbb{Z}_q$

公開鍵： $y = g_1^{x_1} g_2^{x_2}$

メッセージ m に対する署名生成： $r_1, r_2 \in \mathbb{Z}_q$ を選び，$t = g_1^{r_1} g_2^{r_2}$ を計算する．$c = H(g_1||g_2||y||t||m)$ として $s_1 = r_1 - cx_1$, $s_2 = r_2 - cx_2$ を計算する．署名は (c, s_1, s_2) である．

検証： $t' = g_1^{s_1} g_2^{s_2} y^c$ を計算し，$c = H(g_1||g_2||y||t||m)$ となるか検証する．

2.4.2 複数の検証式における SPK

2 つ以上の検証式における SPK も構成可能である．以下に $SPK\{(x_1, x_2) : y = g_1^{x_1} \wedge z = g_1^{x_2} g_2^{x_1}\}(m)$ を示す．

秘密鍵： $x_1, x_2 \in \mathbb{Z}_q$

公開鍵： $y = g_1^{x_1}$，$z = g_1^{x_2} g_2^{x_1}$

メッセージ m に対する署名生成： r_1，$r_2 \in \mathbb{Z}_q$ を選び，$t_y = g_1^{r_1}$，$t_z = g_1^{r_2} g_2^{r_1}$ を計算する．$c = H(g_1||g_2||y||z||t_y||t_z||m)$ として $s_1 = r_1 - cx_1$，$s_2 = r_2 - cx_2$ を計算する．署名は (c, s_1, s_2) である．

検証： $t'_y = g_1^{s_1} y^c$，$t'_z = g_1^{s_2} g_2^{s_1} z^c$ を計算し，$c = H(g_1||g_2||y||z||t'_y||t'_z||m)$ となるか検証する．

3. 従来方式とその問題点

3.1 ブラックリスト型匿名認証システム

ブラックリスト型匿名認証システム [1] は， 2007 年に Tsang らによって初めて提案された．この方式では，前もってユーザの秘密鍵の証明書が資格発行者から発行される．ユーザは証明書を元にチケットと認証データを作成し， SP に対して認証を行う．一方， SP は過去に不正を行ったユーザのチケットを記録したブラックリストを所持しており，ユーザからの認証では，チケットがブラックリストに載っていないことも証明される．この認証では，ゼロ知識証明を用いることにより，認証データは送付するが証明書データ自体は送付されない．そのため， SP に余計な情報を知らせることなく自身が正当なユーザであることが証明できる．そして，不正者排除のために資格発行者を利用することなく，ユーザのプライバシを保護できる．

3.1.1 アルゴリズム

認証時におけるこの方式のアルゴリズムの詳細を以下に示す．SP はユーザに事前に $A^{e+r} = g_0 {g_1}^{sk} {g_2}^y$ を満たす秘密鍵 sk と証明書 (A, e, y) を配布しておく．ここで y は SP の秘密鍵である．

(1) SP はユーザにペア (BL,m) を送信する．ここで，m はランダムチャレンジ，BL= $(\tau_1, ..., \tau_n)$ は SP が保有する現在のブラックリストであり，$\tau_i = (b_i, t_i)$ はブラックリストの i 番目のチケットを表す．また，あるセッションにおいてユーザが不正を行ったとき，SP はそのときのチケットをブラックリストに登録する．

(2) ユーザは自身の秘密鍵 sk を用いて SP にチケット $\tau = (b, t)$ を生成し，SP に送付する．b はセッションを識別する乱数で，タグ t は $t = b^{sk}$ と計算される．なお，SP は DL 仮定より b と t から sk を求めることができないので，ユーザ情報を知ることができない．ユーザはセッションごとに自身の秘密鍵から作られたチケットがブラックリストに入っていないことを証明する必要があるため，不正ユーザは以降認証に失敗する．

(3) ユーザは SP にペア (τ, Π_1) を返す．ここで，τ はユーザ自身のチケット，Π_1 は以下のような SPK である．

$$SPK_1 \left\{ \begin{array}{c} (A, e, sk, y) : A^{e+\gamma} = g_0 g_1^{sk} g_2^y \wedge \\ \left(\bigwedge_{i=1}^n t_i \neq b_i^{sk}\right) \wedge t = b^{sk} \end{array} \right\}(m)$$

SPK_1 により，次の (a)〜(c) が証明される．

(a) $A^{e+\gamma} = g_0 g_1^{sk} g_2^y$ を満たす証明書を保持しており，正規ユーザである．

(b) $\bigwedge_{i=1}^n t_i \neq b_i^{sk}$，つまりユーザは SP のブラックリストに載っていない．

(c) $t = s^{sk}$，つまりチケット τ が正しく生成されている．

(4) ユーザが無効な Π_1 を返した場合は「失敗」，それ以外は「成功」とする．

3.1.2 問題点

上記の方式では，チケットを用いることで資格発行者が不正者を特定することなく，ユーザがブラックリスト化されているか検証できるが，資格発行者が信頼されている前提でないと証明書 A の正当性を保証できない．この問題点に対して，分散ブラックリスト型匿名認証方式 [2] が提案されている．

3.2 分散ブラックリスト型匿名認証システム

ブロックチェーンでの利用を想定して，信頼できる発行者を前提としない分散ブラックリスト型匿名認証システムが， 2019 年に Yang らによって提案されている [2]．ユーザは自身の秘密鍵を用いてチケットを生成する際，自身を含む複数のユーザの公開鍵集合を用いて認証を行う．この公開鍵はブロックチェーン上にあるので，改ざんが難しく，正しさが保証されている．これにより，自身の公開鍵がユーザ集合に含まれることを，自身の公開鍵を明らかにすることなく検証することができるため，資格発行者なし

にチケットの正当性を匿名で保証できる．

3.2.1 アルゴリズム

この方式のアルゴリズムを以下に示す．

- **セットアップ**
  SP が公開パラメータ pp を生成する．
- **登録**
  (1) ユーザ：自身の公開鍵と秘密鍵ペア (pk, sk) を生成する．
  (2) ユーザ：$\Pi_R \leftarrow SPK\{(sk) : T(pk, sk) = 1\}$ を計算することで，証明者が公開鍵 pk に対応する秘密鍵 sk を持っていることを証明できる．ここで T は pk，sk 間の関係を表す．
- **認証**
  (1) ユーザ：SP のサービスにアクセスするためにユーザの公開鍵集合 $\mathcal{C}$ をダウンロードし，正当性を検証する．
  (2) ユーザ：チケット τ と証明 Π_2 を生成し， SP に送る． Π_2 は， $N = |\mathcal{C}|$ に対して以下のような SPK である．

$$SPK_2 \left\{ \begin{array}{c} (sk, pk) : T(pk, sk) = 1 \wedge pk \in \mathcal{C} \wedge \\ \left(\bigwedge_{i=1}^{N} S(sk, \tau_1) \neq 1\right) \wedge S(sk, \tau) = 1 \end{array} \right\} (m)$$

  ここで S はチケット τ，τ_i が sk から生成されていることを示す関係である．このとき，SPK_2 により，次の (a) 〜 (c) が証明される．
  (a) (pk, sk) が正当な鍵ペアである．
  (b) 自身の公開鍵 pk はユーザの公開鍵集合内に含まれる．
  (c) ブラックリスト中の τ_i は sk から生成されておらず，かつ τ は sk から生成されている．
  (3) SP：(τ, Π_2) を受信すると証明 Π_2 を検証し，正当ならば受理する．

3.2.2 問題点

[2] の方式のインスタンス化において， RSA をベースとした場合， RSA アキュムレータを利用することにより効率化している．しかし，短データサイズを実現する楕円曲線暗号ベースでは，単純な OR 型証明しか利用できず，自身の公開鍵がユーザ集合に含まれるかを検証する際に，ユーザの公開鍵集合内の鍵 1 つ 1 つに対して等しいかどうかを検証しなければならない．すなわち，ユーザ数 N に対して認証データサイズが $\mathcal{O}(N)$ となり，効率が悪い．

そこで本研究では，楕円曲線暗号ベースの分散ブラックリスト型匿名認証において，認証データサイズを削減した方式を示す．

4. モデルと安全性の定義

提案方式のモデルと安全性要件を以下に示す．

4.1 モデル

提案方式では，利用者と SP の 2 種類のエンティティとブロックチェーンがあり，以下のプロトコルで構成されている．SP は，セッション ID からチケットを生成し，過去に不正を行ったユーザのチケットを記録したブラックリストを保持している．認証時には，ユーザのチケットとブラックリストのチケットを照合し，ブラックリストに載っていないユーザのみが認証を行うことができる．

- **セットアップ：**システムのセットアップには，システムの公開パラメータを生成するために，信頼できる第三者機関が採用される．このパーティはセットアップの段階でのみ使用され，公開パラメータを素直に生成し，生成プロセスのすべての内部状態を消去することを信頼するだけでよい．
- **登録：**このプロトコルでは，ユーザは自分自身をシステムに登録する．ユーザの正当性は別の手法により保証されているものとする．また，公開鍵はブロックチェーンに登録される．
- **認証：**このプロトコルは，ユーザが匿名で SP のサービスにアクセスする際に，ユーザと SP の間で実行される．ユーザと SP は適切なサイズのユーザ候補の公開鍵集合 (リング)$\mathcal{C}$ に合意する．そして SP はユーザの公開鍵が $\mathcal{C}$ に含まれ，かつチケットがブラックリストに載っていない場合にのみユーザを受け入れる．

4.2 安全性の定義

- **誤認識耐性：**リング $\mathcal{C}$ にある公開鍵に対応した秘密鍵を知らない不正者は認証に成功できない．また，ブラックリストに登録されている不正なユーザも認証に成功できない．
- **匿名性：**SP が認証から知ることができるのは，認証を行ったユーザがブラックリストに登録されているか否かのみであり，ユーザが誰であるかは判断できない．
- **非フレーム性：**あるユーザになりすまして認証に成功して不正を行うことにより，正当なユーザをブラックリストに登録することができない．

5. 提案方式

5.1 提案方式の概要

分散ブラックリスト型匿名認証 [2] において，認証データサイズを削減するために，ブラックリスト型匿名認証システム [1] と効率的なリング署名 [3] を組み合わせた方式を示す．リング署名 [4] とは，自身の秘密鍵を用いて署名を作成する際に，リングと呼ばれる他のユーザの公開鍵集合内に，自身の公開鍵が含まれていることを，公開鍵を明かすことなく示すことにより，匿名で署名を行う方式である．オリジナルのリング署名 [4] では署名長がリングサイ

ズに比例するが， [3] ではリングサイズの log オーダに軽減されている．

ブラックリスト型匿名認証システム [1] では， $pk = g^{sk}$ の形の秘密鍵 sk と公開鍵 pk が利用されており，この sk に基づいてチケットの生成が行われる．一方，効率的なリング署名 [3] の方式でも同じ形の鍵が使われており，その公開鍵 pk がリングに入っていることを，効率化した SPK により証明している．そこで本研究では，[3] のリング署名での SPK を組み込むことによって，ブラックリスト型匿名認証システム [1] におけるユーザがグループに属しているかのチェックを，ユーザ人数が N のとき，認証データサイズを $\mathcal{O}(\log\ N)$ まで削減することが可能となる．また，この方式は， DL 仮定のみに基づいて安全性を示すことができ，ペアリング計算を必要としない．そのため，ペアリングに基づいた従来方式 [1] と比較して，高速に処理を行うことができる．

5.2 提案方式のアルゴリズム

- **セットアップ**
  (1) 位数が大きな素数 q である群 $\mathbb{G}$ を選び，その生成元 g_1 に対して $gk = (\mathbb{G}, q, g_1)$ とする．
  (2) **CGen** を実行し，コミットメント鍵 $ck = (h_1, ..., h_n)$ を出力する．
  (3) $r \leftarrow \mathbb{Z}_q$ をランダムに選び，$ek = {g_1}^r$ を計算する．
  (4) 公開パラメータとして，$pp = (gk, ck, ek)$ を出力する．
- **鍵生成**
  各ユーザは公開パラメータ gk に対して, 秘密鍵 $sk \in \mathbb{Z}_q$ と検証鍵 $vk = g_1^{sk}$ を生成する．vk はブロックチェーン上に登録される．
- **認証**
  (1) ユーザ：秘密鍵 sk を入力として，ρ_3，ρ_4，$b \leftarrow \mathbb{Z}_q$ をランダムに選び, $\beta_3 = \rho_3 \cdot sk$, $\beta_4 = \rho_4 \cdot sk$, $A_3 = g_1^{\rho_3} g_2^{\rho_3}$, $\tilde{A}_1 = (b_1^{sk}/t_1)^{\rho_3}, ..., \tilde{A}_n = (b_n^{sk}/t_n)^{\rho_3}$ を計算する．
  (2) ユーザ：図 3 の SPK を実行することで，自身の公開鍵がリング内に含まれること，自身のチケットが正しく作られていること，自身のチケットがブラックリストに載っていないことを証明する．
- **ブラックリスト管理**
  ブラックリストに追加したいセッションの認証履歴からチケット τ を取り出し，τ をブラックリスト BL に追加する．

5.3 提案方式における SPK

ここでは，提案方式の認証における SPK の詳細を示す．[3] のリング署名での SPK では，暗号文リスト中に 1 の暗号文が存在することを示す SPK を利用しており，その SPK ではさらに複数のビットのコミットの正しさを示す SPK を利用している．提案方式でもこれらを利用する．

- **ビットのコミットであることの証明**
  ビット列から成るメッセージのコミットメントの正しさを示す以下の関係 $\mathcal{R}_1$ の SPK を図 1 に示す．

$$\mathcal{R}_1 = \left\{ \begin{array}{c} (B, (b_{0,0}, ..., b_{m-1,n-1}, r)) : \\ (\forall i, j : b_{j,i} \in \{0,1\}) \wedge (\forall j : \sum_{i=0}^{n-1} b_{j,i} = 1) \wedge \\ B = \mathrm{Com}_{ck}(b_{0,0}, ..., b_{m-1,n-1}; r) \end{array} \right\}$$

  この SPK はリング署名 [4] ベースとなるものであり，図 1 の SPK を実行することで，以下の (1) 〜 (2) が証明できる．
  (1) コミットメント B は $b_{j,1} \in \{0,1\}$ のベクトルから作られている．
  (2) 各 j に対して， 1 つだけ 1 を含む (各ベクトル $(b_{j,0}, ..., b_{j,n-1})$ は 1 つだけ 1 ，他は 0 のベクトルである) ．

  この SPK では, すべての i, j について $b_{j,i}(1-b_{j,i}) = 0$，及び $\sum_{i=1}^{n} b_{j,i} = 1$ を示すことにより，コミットの正しさを示している．
- **リストが 1 の暗号文を含む証明**
  [4] ではさらに，N 個の ElGamal 暗号文のリスト $(c_0, ..., c_{N-1})$ に 1 の暗号文が含まれることを示す以下の関係 $\mathcal{R}_2$ の SPK を利用しており，そのプロトコルを図 2 に示す．

$$\mathcal{R}_2 = \left\{ \begin{array}{c} ((\{c_i\}_{i=0}^{N-1}), (\ell, r)) : (\forall i, c_i \in \mathbb{G}^2) \wedge \\ \ell \in \{0, ..., N-1\} \wedge c_\ell = \mathrm{Enc}_{ck}(1; r) \end{array} \right\}$$

  図 2 について，以下の点に注意する．
  - $N = n^m$ と設定して，m，n に依存した計算を行う．
  - $\ell = i$ のとき $\delta_{\ell,i} = 1$，それ以外のとき $\delta_{\ell,i} = 0$ である．
  - $\delta_{\ell,i} = \prod_{j=0}^{m-1} \delta_{\ell_j,i_j}$ について $\ell = \sum_{j=0}^{m-1} \ell_j n^j$ と $i = \sum_{j=0}^{m-1} i_j n^j$ はそれぞれ ℓ および i の n 進数展開である．
  - 積 $\prod_{j=0}^{m-1} f_{j,i_j}$ は多項式 $p_i(x) = \prod_{j=0}^{m-1} (\delta_{\ell_j,i} x + a_{j,i})$ の x における評価である．
  - $0 \le i \le N-1$ に対して，$p_{i,k}$ は以下の式 (1) が成り立つ係数である．

$$\begin{aligned} p_i(x) &= \prod_{j=0}^{m-1} \delta_{\ell_j,i_j} x + \sum_{k=0}^{m-1} p_{i,k} x^k \\ &= \delta_{\ell,i} x^m + \sum_{k=0}^{m-1} p_{i,k} x^k \end{aligned} \quad (1)$$

  この SPK では, 暗号文 $G_k := \prod_{i=0}^{N-1} c_i^{p_{i,k}} \mathrm{Enc}_{ek}(1; \rho_k)$ を計算して検証者に送ったとき，任意の x について以下の式 (2) が成り立てば c_ℓ は 1 の暗号文であるという性質により，1 の暗号文が含まれることを示している．

$\mathcal{P}_1(gk, ck, ek, B, (b_{0,0}, ..., b_{m-1,n-1}, r))$		$\mathcal{V}_1(gk, ck, ek, B)$
$r_A, r_C, r_D, a_{j,1}, ..., a_{j,n-1} \leftarrow \mathbb{Z}_q$		
$\forall j : a_{j,0} := -\sum_{i=1}^{n-1} a_{j,i}$		
$A := \mathrm{Com}_{ck}(a_{0,0}, ..., a_{m-1,n-1}; r_A)$		
$C := \mathrm{Com}_{ck}(\{a_{j,i}(1-2b_{j,i})\}_{j,i=0}^{m-1,n-1}; r_C)$	A, C, D $\longrightarrow$	Accept if and only if
$D := \mathrm{Com}_{ck}(-a_{0,0}^2, ..., -a_{m-1,n-1}^2; r_D)$		$A, B, C, D \in \mathbb{G}$
	$x \leftarrow \{0,1\}^\lambda$	$f_{0,1}, ..., f_{m-1,n-1}, z_A, z_C \in \mathbb{Z}_q$
$\forall j, i : f_{j,i} := b_{j,i} x + a_{j,i}$	$\longleftarrow$	$\forall j : f_{j,0} := x - \sum_{i=1}^{n-1} f_{j,i}$
$z_A := rx + r_A$	$f_{0,1}, f_{1,1}, ..., f_{m-1,n-1}, z_A, z_C$	$B^x A = \mathrm{Com}_{ck}(f_{0,0}, ..., f_{m-1,n-1}; z_A)$
$z_C := r_C x + r_D$	$\longrightarrow$	$C^x D = \mathrm{Com}_{ck}(\{f_{j,i}(x - f_{j,i})\}_{j,i=0}^{m-1,n-1}; z_C)$

図 1 $\mathcal{R}_1$ の SPK

$\mathcal{P}_2(gk, ck, ek, (c_0, ..., c_{N-1}), (\ell, r))$		$\mathcal{V}_2(gk, ck, ek, (c_0, ..., c_{N-1}))$
$r_B, \rho_k \leftarrow \mathbb{Z}_q$		
$B := \mathrm{Com}_{ck}(\delta_{l_{0,0}}, ..., \delta_{l_{m-1,n-1}}; r_B)$		
$(A, C, D) \leftarrow \mathcal{P}_1(gk, ck, ek, B, (\{\delta_{\ell_j,i}\}_{j,i=0}^{m-1,n-1}, r_B))$		
For $k = 0, ..., m-1$		
$G_k = \prod_{i=0}^{N-1} c_i^{p_{i,k}} \cdot \mathrm{Enc}_{ek}(1; \rho_k)$	$A, B, C, D, \{G_k\}_{k=0}^{m-1}$	
using $p_{i,k}$ from(1)	$\longrightarrow$	Accept if and only if
	$x \leftarrow \{0,1\}^\lambda$	$A, B, C, D, G_0, ..., G_{m-1} \in \mathbb{G}$
$(f_{0,1}, ..., f_{m-1,n-1}, z_A, z_C) \leftarrow \mathcal{P}_1(x)$	$\longleftarrow$	$f_{0,1}, ..., f_{m-1,n-1}, z_A, z_C, z \in \mathbb{Z}_q$
$z := rx^m - \sum_{k=0}^{m-1} \rho_k x^k$	$f_{0,1}, ..., f_{m-1,n-1}, z_A, z_C, z$	$\mathcal{V}_1(gk, ck, ek, B, x, A, C, D, \{f_{j,i}\}_{j=0,i=1}^{m-1,n-1}, z_A, z_C) = 1$
	$\longrightarrow$	$\prod_{i=0}^{N-1} c_i^{\prod_{j=1}^{m} f_{j,i_j}} \cdot \prod_{k=0}^{m-1} G_k^{-x^k} = \mathrm{Enc}_{ek}(1; z)$

図 2 $\mathcal{R}_2$ の SPK

$$\prod_{i=0}^{N-1} c_i^{\prod_{j=0}^{m-1} f_{j,i_j}} \cdot \prod_{k=0}^{m-1} G_k^{-x^k} = \left(\prod_{i=0}^{N-1} c_i^{\delta_{\ell,i}} \right)^{x_m} \quad (2)$$

- **提案方式の認証における SPK**

以下の関係 $\mathcal{R}_{\mathrm{auth}}$ を示す提案方式の認証における SPK を図 3 に示す.

$$\mathcal{R}_{\mathrm{auth}} = \left\{ \begin{array}{c} ((m, \mathcal{C}), sk) : sk \in \mathbb{Z}_q \wedge \\ vk = {g_1}^{sk} \in R \subset \mathbb{G}^* \wedge A_3 = {g_1}^{\rho_3} {g_2}^{\rho_4} \wedge \\ 1 = {A_3}^{-sk} {g_1}^{\beta_3} {g_2}^{\beta_4} \wedge \\ \tilde{A}_1 = ({b_1}^{sk}/t_1)^{\rho_3}, ..., \tilde{A}_n = ({b_n}^{sk}/t_n)^{\rho_3} \wedge \\ t = b^{sk} \end{array} \right\}$$

図 3 の SPK を実行することで，以下の (1)〜(3) が証明できる.

(1) ユーザ自身の公開鍵がリング $\mathcal{C}$(ユーザの公開鍵集合) に含まれる.

(2) 自身のチケットがブラックリストに載っていない.

(3) 自身のチケット τ が秘密鍵から正しく作られている.

ここで，(1) は効率的なリング署名 [3] ，(2) 及び (3) はブラックリスト型匿名認証 [1] の方式に基づいている．図 3 について，効率的なリング署名のアルゴリズムを用いた部分を説明する.

(a) 自身の公開鍵 pk を暗号化した値 d を生成する.

(b) 暗号文 d をリング内のすべての公開鍵 $pk_0, ..., pk_{N-1}$ の暗号文で除した暗号文 $c_0, ..., c_{N-1}$ を生成する.

(c) $c_0, ..., c_{N-1}$ を $\mathcal{P}_2$ に送ることで，リスト $c_0, ..., c_{N-1}$ について 1 つだけ 1 の暗号文がある，即ち自身の公開鍵から生成された暗号文が 1 つだけ存在することを証明する．(この証明は $\mathcal{P}_{\mathrm{sig}} \rightarrow \mathcal{P}_2 \rightarrow \mathcal{P}_1$ の入れ子構造になっている．)

(a) 〜 (c) のアルゴリズムにより，公開鍵集合内に自身の公開鍵が 1 つだけ含まれることを証明することができる.

5.4 提案方式の安全性

提案方式の安全性は以下のような観点から達成できている.

- **誤認識耐性：**図 3 の SPK では，(1)〜(3) が証明されており，SPK の安全性 (soundness) から公開鍵がリングに含まれない，もしくはブラックリストにチケットが含まれるユーザは正しい SPK を生成できないため成り立つ.
- **匿名性：**認証で SP へ送信される情報が SPK とコミットメントのみのため，ユーザを特定する情報が得られ

$\mathcal{P}_{\text{auth}}(gk, ck, ek, (m, \mathcal{C}), sk)$		$\mathcal{V}_{\text{auth}}(gk, ck, ek, (m, \mathcal{C}, c))$
$s, \zeta, \xi, \rho_3, \rho_4, r_{\rho_3}, r_{\rho_4}, r_{\beta_3}, r_{\beta_4} \leftarrow \mathbb{Z}_q$		
$d \leftarrow \text{Enc}_{ek}(g_1^{sk}; \zeta) = (ek^{\zeta}, g_1^{\zeta} g_1^{sk})$		
$B \leftarrow \text{Enc}_{ek}(g_1^{s}; \xi) = (ek^{\xi}, g_1^{\xi} g_1^{s})$		
$c_0 := d \cdot \text{Enc}_{ek}(pk_0^{-1}; 0), ..., c_{N-1} := d \cdot \text{Enc}_{ek}(pk_{N-1}^{-1}; 0)$		
$= (ek^{\zeta}, g_1^{\zeta} g_1^{sk} pk_0^{-1}), ..., (ek^{\zeta}, g_1^{\zeta} g_1^{sk} pk_{N-1}^{-1})$		
$a_2 \leftarrow \mathcal{P}_2(gk, ck, ek, (c_0, ..., c_{N-1}), (\ell, \zeta))$		Accept iff
$t_1 = {g_1}^{r_{\rho_3}} {g_2}^{r_{\rho_4}}$		$\mathcal{C} \subset \mathbb{G}^*; \quad d, B \in \mathbb{G}^2$
$t_2 = A_3^{-s} {g_1}^{r_{\beta_3}} {g_2}^{r_{\beta_4}}$		$z_s, z_\xi \in \mathbb{Z}_q$
For $\quad i = 1, ..., n$	$d, B, a_2,$	$d^x B = \text{Enc}_{ek}({g_1}^{z_s}; z_\xi)$
$\quad t_{3i} = {b_i}^{r_{\beta_3}} {t_i}^{-r_{\rho_3}}$	$t_1, t_2, \{t_{3i}\}_{i=1,...,n}, t_4$	$\quad = (ek^{z_\xi}, {g_1}^{z_\xi} {g_1}^{z_s})$
$t_4 = b^{r_{\beta_3}} t^{-r_{\rho_3}}$	$\longrightarrow$	$\mathcal{V}_2(gk, ck, ek, (c_0, ..., c_{N-1}), a_2, x, z_2) = 1$
	$x \leftarrow \{0,1\}^{\lambda}$	$t_1 = {g_1}^{s_{\rho_3}} {g_2}^{s_{\rho_4}} {A_3}^{-x}$
$z_s := sk \cdot x + s; \quad z_\xi := \zeta x + \xi$	$\longleftarrow$	$t_2 = {A_3}^{-z_s} {g_1}^{s_{\beta_3}} {g_2}^{s_{\beta_4}} (1)^{-x}$
$z_2 \leftarrow \mathcal{P}_2(x)$	$z_s, z_\xi, z_2,$	For $\quad i = 1, ..., n$
$s_{\beta_3} := \beta_3 x + r_{\beta_3}; \quad s_{\rho_3} := \rho_3 x + r_{\rho_3}$	$s_{\rho_3}, s_{\rho_4}, s_{\beta_3}, s_{\beta_4}$	$\quad t_{3i} = {b_i}^{s_{\beta_3}} {t_i}^{-s_{\rho_3}} \tilde{A}_i^{-x}$
$s_{\beta_4} := \beta_4 x + r_{\beta_4}; \quad s_{\rho_4} := \rho_4 x + r_{\rho_4}$	$\longrightarrow$	$t_4 = b^{s_{\beta_3}} t^{-s_{\rho_3}} (1)^{-x}$

図 3 $\mathcal{R}_{\text{auth}}$ の SPK

ない.

- **非フレーム性：**正当なユーザになりすますためにはそのユーザの sk が必要になるが，SPK から sk は漏れないため，なりすますことができない.

6. 実装による計測結果と考察

6.1 実装環境と評価方法

提案方式の有効性を検証するために，表 1 に示す実装環境で提案方式を実装した.

表 1 実装環境

OS	Ubuntu 20.04.1 LTS (64bit)
CPU	Intel ® Core™ i5-7400 (3.00GHz)
メモリ	7.7GiB
プログラミング言語	C 言語
ライブラリ	OpenSSL 1.1.1f

提案方式では，双線形写像を使用しておらず，楕円曲線上における DL 仮定のみに依存しているため，ペアリングライブラリではなく，OpenSSL の楕円曲線暗号ライブラリを使用した.

ただし，今回の実装では，図 3 の $p_{i,k}$ を式 (1) で生成される値はなく，乱数を用いている.

6.2 測定結果と評価

6.2.1 データサイズの比較

図 4 において，ブラックリストサイズを 100 としたときの，従来方式 [2] および提案方式におけるユーザ数と認証データサイズ (バイト) の関係を示す．効率的なリング署名と組み合わせることで log オーダサイズを実現しており，従来方式よりも認証データサイズを大幅に削減できている.

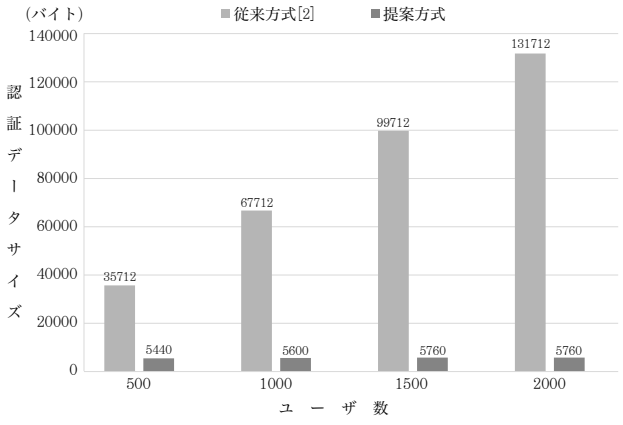

図 4 ユーザ数と認証データサイズの関係

なお，従来方式における認証データサイズのうちの 32 バイト，提案方式における認証データサイズのうちの 96 バイトは検証者から証明者への認証データである.

6.2.2 処理時間の評価

- **ユーザ数を変化させた場合**

図 5 のグラフでは，ブラックリストのサイズを 100 に固定し，ユーザ数 N を変化させたときの，提案方式における $\mathcal{P}_{\text{auth}}$($\mathcal{P}_1$ と $\mathcal{P}_2$ を含む) および $\mathcal{V}_{\text{auth}}$($\mathcal{V}_1$ と $\mathcal{V}_2$ を含む) の処理時間でを示している．$\mathcal{P}_2$ の SPK にお

いて，G_k を求める際に mN 回計算を行っているため，$\mathcal{V}_{\text{auth}}$ よりも $\mathcal{P}_{\text{auth}}$ の方が処理時間がかかっていることが分かる．また，$\mathcal{P}_{\text{auth}}$，$\mathcal{V}_{\text{auth}}$ とも，N 回計算を行っている部分があるため，検証の処理時間もユーザ数 N に比例して増加している．よって，認証に用いるユーザの公開鍵集合が多いほど，匿名性は向上するが，計算時間も増大することが分かる．ただし，ブロックチェーンのアプリケーションでは，ブロックチェーン上での検証の高速化が重要であり，ユーザ数 2000 程度でも実用的な時間に収まっている．

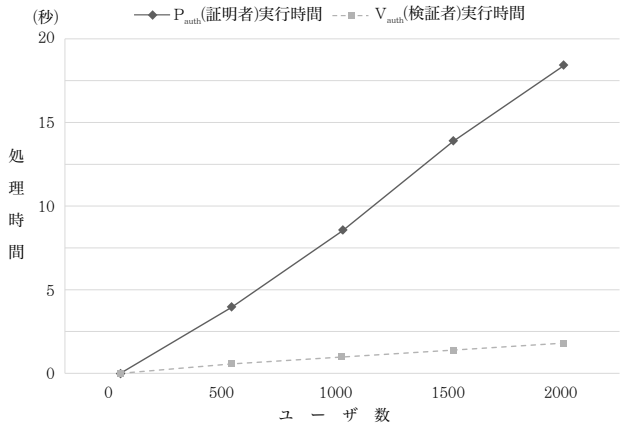

図 5 ユーザ数と処理時間の関係

- **ブラックリストサイズを変化させた場合**

  図 6 のグラフでは，ユーザ数を 1000 に固定し，ブラックリストサイズを変化させたときの，提案方式における $\mathcal{P}_{\text{auth}}$($\mathcal{P}_1$ と $\mathcal{P}_2$ を含む) および $\mathcal{V}_{\text{auth}}$($\mathcal{V}_1$ と $\mathcal{V}_2$ を含む) の処理時間を示す．$\mathcal{P}_{\text{auth}}$，$\mathcal{V}_{\text{auth}}$ とも，処理時間はブラックリストサイズに比例してわずかに増加しているが，提案方式の処理時間は実用的であるといえる．

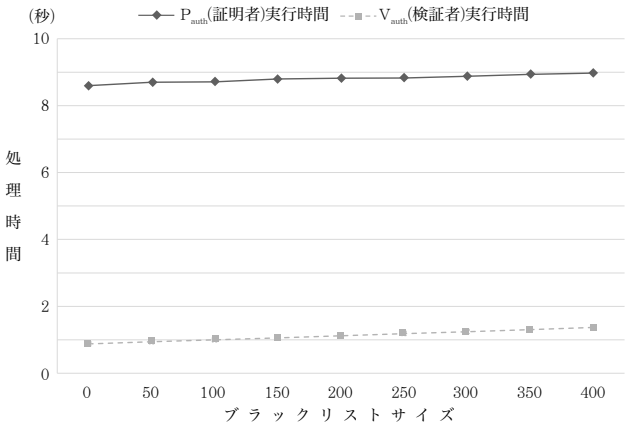

図 6 ブラックリストサイズと処理時間の関係

7. まとめ

本研究では，従来のブラックリスト型匿名認証システムと効率的なリング署名を組み合わせた分散ブラックリスト型匿名認証システムを提案した．従来方式では，ユーザの正当性を保証するために，ユーザ集合内すべての公開鍵に対して検証を行う必要があり，認証データサイズがユーザ数に比例してしまう．そこで，効率的なリング署名を用いて検証を行うことで，認証データサイズを log オーダに削減した．

また，PC 上に実装し測定した結果から，ブラックリストサイズが増大しても，全体の処理時間には大きな影響がないことを確認した．一方，認証に用いるユーザ数が増大すると，処理時間はユーザ数に比例して増加してしまうことが分かった．そこで今後の課題としては，ユーザ数が変化しても処理時間に大きな影響を与えない方式の提案などが考えられる．

参考文献

[1] P. P. Tsang, M. H. Au, A. Kapadia, and S. W.Smith, “Blacklistable Anonymous Credentials: Blocking Misbehaving Users without TTPs”, ACM-CCS2007, pp. 72-81, 2007.

[2] R. Yang, M. H. Au, Q. Xu, and Z. Yu, “Decentralized Blacklistable Anonymous Credentials with Reputation”, In Computers & Security, Vol.85, pp.353-371, 2019.

[3] J. Bootle, A. Cerulli, P. Chaidos, E. Ghadafi, J. Groth, and C. Petit, “Short Accountable Ring Signatures Based on DDH”, ESORICS2015, LNCS9326, pp.243-265, 2015.

[4] R. L. Rivest, A. Shamir, and Y. Tauman, “How to leak a secret”, ASIACRYPT2001, LNCS2248, pp.552-565, 2001.

[5] T. ElGamal, “A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms”, CRYPTO’84, LNCS196, pp.10-18, 1984.

[6] T. P. Pedersen, “Non-Interactive and Information-Theoretic Secure Verifiable Secret Sharing”, CRYPTO’91, LNCS576, pp.129-140, 1991.