失効可能グループ署名における ブルームフィルタを用いた失効リストの削減

木村 魁[1,a)]　中西 透[1,b)]

概要：グループ署名とは，あるグループに所属するユーザが匿名性を保ちつつ「グループに所属している」ことを証明できる署名方式である．グループ署名の活用例として Crowdsensing があり，ユーザの位置情報を取り扱うためグループ署名を用いたユーザ認証によりプライバシーが保護される. グループ署名では，失効されたユーザからの送信を匿名のまま防ぐために失効機能が必要となる．先行研究では，Crowdsensing 向けの効率的な失効可能グループ署名が提案されている．しかし，この方式では定期的に失効者リストを検証者に送信する必要があるが，送付するデータサイズが大きく，検証者が分散している場合問題となる. そこで本研究では，確率的データ構造である Bloom filter を用いることで，失効リストのデータサイズを削減することを目的とする．Bloom filter には偽陽性がありその確率は制御可能である．失効タグの数，求められる偽陽性の発生確率を考慮し Bloom filter を適用した上で，失効リストのデータ削減度ならびに通信時間短縮の評価を行う.

キーワード：グループ署名，失効，Bloom filter，プライバシー保護

Reducing Revocation List in a Revocable Group Signature Scheme Using Bloom Filter

Kai Kimura[1,a)]　Toru Nakanishi[1,b)]

Abstract: A group signature is a signature scheme that allows a user in a group to prove the membership to the group while keeping the anonymity. Group signatures require a revocation function to prevent revoked users from sending messages. Previously, a revocable group signature scheme for crowdsensing has been proposed. However, this scheme requires periodic transmission of the revocation list to the verifiers, and the data size becomes a problem. In this study, we aim to reduce the data size of the revocation list by using Bloom filter, where the Bloom filter is a probabilistic data structure with false positives, but its probability can be controlled. Considering the number of revoked users, the acceptable probability of false positives, we applied the Bloom filter, and then we evaluated the degree of reduction of data size and the communication time.

Keywords: group signatures, revocation, Bloom filter, privacy protection

1. はじめに

グループ署名 [1][2] とは，あるグループに所属するユーザが秘匿性を保ちつつ「グループに所属している」ことを証明できる署名方式である．この方式ではグループ管理者 (Group Manager) と呼ばれる信頼できる機関がユーザのグループへの加入や脱退を管理している．通常のユーザ認証では「ID」と「パスワード」等の認証情報をサーバへ送り，認証を行う．サーバは ID を紐付けすることにより，そのユーザのアクセス履歴を収集できるため，これらの情報

[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)] m214825@hiroshima-u.ac.jp
[b)] t-nakanishi@hiroshima-u.ac.jp

が他者へ漏洩するリスクがある．これに対してグループ署名では，ユーザ個人を認証するのではなく，所属しているグループを認証する．この際ユーザ ID は秘匿され，個々のユーザを識別することができない．グループ署名の活用例として，Crowdsensing[3] が挙げられる．Crowdsensing は Mobile Crowdsensing と呼ばれることもあり，スマートフォンなどのセンサを持つモバイル機器を保有している多数のユーザーが移動しながらセンシングを行い，サーバはその収集，分析を行う手法である．Crowdsensing では，多数の人々の位置情報を扱うため，サーバにおいて漏洩するリスクがあるが，グループ署名を用いた匿名認証によって，その問題が解決される．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からの通信を拒否するために，失効機能が必要となる．Crowdsensing 向けの失効可能グループ署名として，SRBE[4], GS-TDL[5], Sucasas らの方式 [6] が提案されている．これらの方式は VLR(Verifier-Local Revocation) 型になっており，サーバ側で失効の処理が完結されるため，クライアント（ユーザ）側の負担が少ない．従来の VLR 方式は検証時間が失効数に比例した暗号演算を必要としていたが，これらの方式では各署名中と失効リスト中の失効タグをマッチングすることにより，検証時の失効確認が高速で行える．これらの方式では，時間を一定期間毎に区切り，失効タグは生成される期間 T を基に生成される．

しかし，SRBE[4], GS-TDL[5] では, 失効タグが T のみに依存して生成されるため，同一期間内に生成された失効タグは同一のものとなる．よって，同一期間内では，2 つの署名が同一署名者であることが判明する (Linkability を持つ)．一般的に署名がリンク可能であることは匿名性の観点から問題がある．また，Sucasas らの方式 [6] においては，失効タグ τ は T に加え，タスク ID と呼ばれる各ユーザのタスクに割り当てられた ID に基づいて作成される．タスク ID を毎回変えることにより，同期間内において同一の失効タグが生成されることは無い (linkability を持たない) が，タスク ID が検証者に公開されているため，匿名性に問題がある．先行研究 [7] では，以上の問題点を踏まえた上で，匿名性を強化した Crowdsensing 向け失効可能グループ署名が提案されている．この先行研究の方式では，同一期間内における署名の生成回数の上限 Δ，ならびに同一期間内における現在の署名生成回数を示すパラメータ $k(1 \leq k \leq \Delta)$ を導入し，失効タグ τ が T, k に依存して生成されるため，各署名がリンク不能になっている．さらに，k を暗号化し k に対する署名をゼロ知識証明することで，k の値の漏えいを防いでいる．しかし，先行研究の手法では期間が更新される度，全失効者に対し Δ 個の失効タグを計算し，各検証者へ送信する必要がある．この失効タグが格納される失効リストのデータサイズは，失効者数，ならびに Δ に比例して増加し，失効者数 1 万人，$\Delta = 100$ の際，61MB となる．自動運転に代表される路車間，車車間通信等，検証者が分散しているケースを想定すると，この失効リストサイズでは問題となる．

そこで本研究では，確率的データ構造である Bloom filter を用いることで，失効リストのデータサイズを削減することを目的とする．Bloom filter は他の集合的データ構造と比較した際，非常に空間効率が良くストレージを節約することが可能である．また，要素の登録や検索に必要とする処理時間が $\mathcal{O}(1)$ であり，高速に処理を行うことができる．特徴として，Bloom filter は偽陽性が発生することが挙げられる．Crowdsensing 上での偽陽性の発生とは，「正規ユーザが失効されている」と誤って検出されることであり，逆に悪意を持つような失効されたユーザが認証を通過することは不可能である．その為，安全性は保証される．また，偽陽性の発生確率が 1%の場合，必要なデータは 1 要素あたり 9.6bit であり，その発生確率は，登録する要素数や，用意する Bloom filter の数を調整することで制御可能である．そこで本研究では，失効タグの総数，求められる偽陽性の発生確率等を考慮し，失効リストのデータ削減度，ならびに通信時間短縮等の評価を行う．

2. 数学的準備

2.1 双線形写像

先行研究 [7] の方式では，双線形写像が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 p の巡回群とする．また，$\mathbb{G}_1, \mathbb{G}_2$ それぞれの生成元を 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 $u \in \mathbb{G}_1, v \in \mathbb{G}_2, a, b \in \mathbb{Z}_p$ について $e(u^a, v^b) = e(u, v)^{ab}$
- **非退化性：**$e(g, h) \neq 1_{\mathbb{G}_T}$

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可能である。

2.2 知識の署名

知識の署名 (SPK: Signature based on Proofs of Knowledge) は，知識のゼロ知識証明をハッシュ関数を用いて非対話型に変換することで得られる署名である．知識のゼロ知識証明とは証明者 P と検証者 V と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．

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

2.3 Bloom filter

本研究では，さらに Bloom filter を用いて，失効リストサイズの削減を行う．Bloom filter は 1970 年に Burton

Howard Bloom によって考案された空間効率の良い確率的データ構造である．ある要素がその集合に含まれているか否かの判定に用いられる. 同データ構造は偽陽性が発生するという特性があるが、その確率が制御されている場合，大幅な空間の節約が期待できる．偽陽性とは，実際には含まれていない要素が「含まれている」と誤判定されることである．また，要素の追加は可能であるが，削除は不可能であり，要素を追加するたびに偽陽性の発生確率は上昇する．以下に，同データ構造への要素の追加を行う登録アルゴリズム，ならびに要素の有無を確認する検索アルゴリズムを示す．また，図 1 に $m = 18, k = 3$ の例を示す．

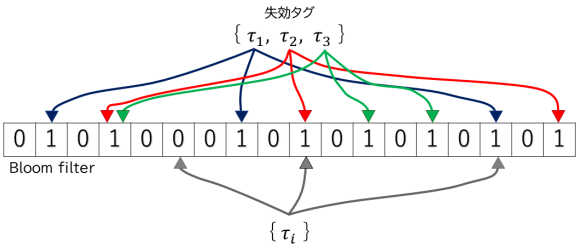

図 1 Bloom filter の例 ($m = 18,\ k = 3$)

- 登録
  (1) サイズ m ビットの配列を用意し，配列の各ビットを'0' に初期化する．
  (2) 登録したいビット列の要素を 0 〜 $m-1$ の値を出力する k 個のハッシュ関数に適用する．
  (3) 出力された k 個のハッシュ値のインデックスの配列の bit を '1' とする．
- 検索
  (1) 検索したい要素を同一のハッシュ関数 k 個に適用する．
  (2) 出力された k 個のハッシュ値のインデックスの配列の bit 値を確認する．
  (3) 確認した配列の bit 値中において，以下のように判定する．
    (a) 1bit 以上 '0' が含まれている場合：要素は集合に含まれていない．
    (b) 全ての bit が '1' であった場合： 要素は集合に含まれている，もしくは偶然全ての bit が '1' となっており，偽陽性が発生したと考えられる．
- 偽陽性確率
  以下に，偽陽性の発生確率を示す．
  偽陽性が発生する要因は，ハッシュ関数によって計算された k 個の配列位置のビットが偶然全て '1' となっていることであるので，偽陽性確率 ε は以下のようになる．

$$\varepsilon = \left(1 - \left[1 - \frac{1}{m}\right]^{kn}\right)^k \approx \left(1 - e^{-\frac{kn}{m}}\right)^k$$

- 最適なハッシュ関数の数と 1 要素あたりに必要なデータ量

  ハッシュ関数の数 k, 与えられた要素の数 n と Bloom filter のサイズ m に対して，偽陽性の確率を最小化する k の値は

$$k = \frac{m}{n} \ln 2$$

  である．このとき，ε に対して一要素あたりに必要な最適な bit 数は，

$$\frac{m}{n} = -\frac{\log_2 \varepsilon}{\ln 2} \approx -1.44 \log_2 \varepsilon$$

  となる．

3. 先行研究

3.1 先行研究の概要

先行研究 [7] の方式では，従来方式 [4][5][6] と同様にマッチングによる効率的な失効チェックが可能である．さらに，期間内の署名もリンク不可能であると共に，Sucasas の方式 [6] の匿名性の問題も改善されている．Scasas の方式では，単位期間内における署名の生成回数を表すタスク ID がそのまま署名に含まれており，それが第三者に傍受された場合，匿名性が破られる可能性がある．この問題を解決するため，先行研究 [7] の方式では，署名中にタスク ID の代替として期間内の署名作成回数 k を採用し，その k をそのまま署名に埋め込まず，匿名性を強化している．

この [7] の方式では，単位期間において生成可能な署名の生成回数の上限 Δ が定められている．失効タグは，ユーザの秘密鍵 x，期間 T と署名回数 $k(1 \le k \le \Delta)$ に基づいて作られているため，各署名がリンク不能である．k の値は 1〜Δ とならねばならないが，署名中に含まれていないためサーバ側は k がこの範囲内の値であるか否かが確認できない．この問題を解決するために，k に対する署名をゼロ知識証明することによって，k の正当性が保証されている．

3.2 先行研究での失効処理

先行研究 [7] における失効処理は，GS-TDL[5] の失効タグを拡張する形で実現されている．ここでは，本研究と関連がある箇所について簡単に紹介する．失効されたユーザの失効タグ τ を失効リスト RL に格納すると共に，署名中の失効タグ τ とマッチングで高速に照合を行い正規のユーザが否かを判断する．失効タグ τ は以下のように生成される．

$$\tau = g_1^{\frac{1}{x+k+T}}$$

ここで x はユーザが保持する秘密情報，k が単位期間内

における現在の署名生成回数，T はその期間を表している（ただし，$k+T$ は取りうる値が重複すると，同一の失効タグが生成されてしまうため，重複しないように設定される）．x, k, T に依存して生成されることにより，全期間における全失効者に対する全失効タグが区別可能である．

3.3 先行研究の問題点

先行研究の方式では，1 人の失効者に対して一定期間毎に Δ 回分の失効タグを生成する必要がある．よって，検証者へ配布する失効リストが，失効者数ならびに単位期間中に生成可能な署名の上限数 Δ に比例し，サイズが問題となる．失効タグのデータサイズは 1 つあたり 64byte(512bit) であり，失効者数 10,000 人，Δ=100 の場合，失効リストのサイズは 61MB となる．自動運転に代表される路車間，車車間通信等，検証者が分散しているケースを想定すると問題となる．

4. 提案方式

4.1 提案方式の概要

本研究では，確率的データ構造である Bloom filter を用いて，先行研究 [7] の方式における失効リストのデータ削減を行う．Bloom filter に失効タグを格納して失効リストとすることにより，大幅にデータ削減が可能である．このとき，偽陽性の確率を考慮して，Bloom filter のサイズを設定する必要がある．失効リスト作成時には，ベースの方式 [1] に従い全ての失効ユーザに対して Δ 個の失効トークンを作成し，それらを Bloom filter に格納して失効リストとして配布する． 失効されていないユーザであることを確認するには，配布された Bloom filter に対し，グループ署名中のタグが Bloom filter に登録されていないことを確認する．その際，偽陽性が発生する恐れがあるが，失効者数や偽陽性の発生確率の許容範囲などを考慮し，最適なサイズの Bloom filter を適用させる. その評価は 5 章で示す.

4.2 各アルゴリズムの定義

提案方式は以下の **KeyGen**, **Join**, **GSign**, **Verify**, **Revoke**, **Open** の 6 つのアルゴリズムから成る．

- **KeyGen** : セキュリティパラメータ λ と各期間での署名作成回数の上限 Δ を入力とする．グループ公開鍵 gpk，グループ秘密鍵 gsk，初期化したユーザリスト $u\_list := \phi$，初期化した Bloom filter による失効リスト $BF_0 := \phi$ を出力する．
- **Join** : gpk, gsk, u_list, グループに加入するユーザの ID を入力とする．ID を持つユーザの署名鍵 $sigk_{ID}$ を出力し，u_list の更新を行う．
- **GSign** : gpk，期間 T，$sigk_{ID}$，署名されるメッセージ M，ユーザの現在の署名作成回数 k を入力とする．グループ署名 σ を出力する．
- **Verify** : gpk, 期間 T における失効リスト BF_T, M, σ, T を入力とする．出力は 1（**valid**）または 0（**invalid**）のどちらかである．
- **Revoke** : gpk，u_list，ID，T，BF_T を入力とする．ID を持つユーザの失効タグをハッシュ関数に通し，Bloom filter を更新する．期間 T が更新されたときは，失効ユーザ全てに対して Bloom filter の更新を行う．
- **Open** : u_list，σ，M，T を入力とする．出力は σ の署名者の ID または 0 となる．

4.3 提案方式のアルゴリズム

以下に，提案方式のアルゴリズムを示す．先行研究 [7] より，変更されたアルゴリズムは次の 2 つである．まず，署名検証で失効確認を行う **BfRevCheck** である．先行研究 [7] では，失効ユーザの失効タグから成る失効リストに対し署名中の失効タグのマッチング検索を行っていた．提案手法では，この失効リストの代わりに失効タグを格納した Bloom filter に対して，検索を行う．また，**Revoke** も変更されている．先行研究 [7] では，期間 T が更新されるたびに，全ての失効ユーザに対する失効タグを生成し，失効リストとしている．提案方式では，その生成された失効タグに対して Bloom filter を更新する．他の， **KeyGen, Join, GSign, SignCheck, Open** アルゴリズムに変更点はない.

- **KeyGen**$(1^\lambda, \Delta)$: 与えられたセキュリティパラメータ λ に対して λ ビットの素数位数 p を持つ双線形群 $(\mathbb{G}_1, \mathbb{G}_2)$ を選ぶ．$\mathbb{G}_1, \mathbb{G}_2$ 上の生成元を g_1, g_2 とする．GM は $\gamma \xleftarrow{R} \mathbb{Z}_p$ と $h, g_1' \xleftarrow{R} \mathbb{G}_1$ を選び，$W = g_2^\gamma$ を計算する．グループ公開鍵 $gpk = (g_1, g_2, h, g_1', W, \Delta)$, グループ秘密鍵 $gsk = \gamma$, ユーザリスト $u\_list := \phi$, 失効リスト $RL_0 = \phi$ を出力する．
- **Join**(gsk, u_list, ID) : ユーザ ID の秘密情報を $x, y \xleftarrow{R} \mathbb{Z}_p$ とする．すべての署名回数 $k(1 \leqq k \leqq \Delta)$ に対して署名 $A_k = (g_1 g_1'^k h^{-y})^{\frac{1}{\gamma+x}}$ を計算する．
  署名鍵 $sigk_{ID} = (x, y, \{A_k\}_{k=1,\ldots,\Delta})$ を出力し，$u\_list := u\_list \cup \{(ID, x)\}$ として u_list の更新をする．
- **GSign**$(gpk, T, sigk_{ID}, M, k)$: $sigk_{ID} = (x, y, \{A_k\}_{k=1,\ldots,\Delta})$ とする．$\beta \xleftarrow{R} \mathbb{Z}_p, \delta = \beta x - y$ とし，k の署名 A_k のコミットメント $C = A_k h^\beta$, 失効タグ $\tau = g_1^{\frac{1}{x+k+T}}$ を計算する．$r_x, r_\delta, r_\beta, r_k \xleftarrow{R} \mathbb{Z}_p$ として，SPK を以下のように計算する．

$$R_1 = \frac{e(h,g_2)^{r_\delta} e(h,W)^{r_\beta} e(g_1',g_2)^{r_k}}{e(C,g_2)^{r_x}},$$
$$R_2 = e(\tau,g_2)^{r_x+r_k},$$
$$c = H(gpk, C, \tau, R_1, R_2, M),$$
$$s_x = r_x + cx,\ s_\delta = r_\delta + c\delta,$$
$$s_\beta = r_\beta + c\beta,\ s_k = r_k + ck$$

この SPK はある θ, ξ に対して，

$$e(A_k, g_2^{x+\gamma}) = e(g_1 g_1'^k h^\theta, g_2) \quad (1)$$
$$e(\tau, g_2^{x+k+T}) = e(g_1, g_2) \quad (2)$$

すなわち署名 A_k と失効タグ τ の正しさを証明している． グループ署名 $\sigma = (C, \tau, c, s_x, s_\delta, s_\beta, s_k)$ を出力する．

- **Verify**$(gpk, RL_T, M, \sigma, T)$: 以下の 2 つのサブアルゴリズムが $valid$ を出力したときにこのアルゴリズムは $valid$ を出力する．
  - **SignCheck**(gpk, T, σ, M) : SPK を検証するために，以下の式を計算する．

$$R_1' = \frac{e(h,g_2)^{s_\delta} e(h,W)^{s_\beta} e(g_1',g_2)^{s_k}}{e(C,g_2)^{s_x}} \left(\frac{e(C,W)}{e(g_1,g_2)}\right)^{-c}$$
$$R_2' = e(\tau,g_2)^{s_x+s_k} \left(\frac{e(g_1,g_2)}{e(\tau,g_2^T)}\right)^{-c}$$

もし $c = H(gpk, C, \tau, R_1', R_2', M)$ なら $valid$ を，そうでなければ $invalid$ を出力する．

  - **BfRevCheck**(T, BF_T, σ) **：**
署名 σ 中の τ を k 個のハッシュ関数に入力し，k 個の配列インデックスを得る．それらの位置の配列のビット値に 0 が 1 つ以上含まれていた場合，その要素は Bloom filter に登録されていないことが示される．一方，全ての位置において 1 であった場合は，その要素は Bloom filter に登録されている，もしくは偽陽性が発生したと考えられる．1 つ以上 0 が含まれていた場合 $valid$, そうでなければ $invalid$ を出力する．

- **Revoke**$(gpk, u\_list, ID, T, BF_T)$:
u_list から ID に対応する x を得て，$\{g_1^{\frac{1}{x+1+T}}, \ldots, g_1^{\frac{1}{x+\Delta+T}}\}$ の値を計算し，それぞれ k 個のハッシュ関数に通し，得られた位置の列を Bloom filter に登録する．また，期間が更新された際は，それまでに失効された全ユーザに対してもう一度 Bloom filter を作り直す．

- **Open**(u_list, σ, M, T) : σ 中から τ を得る．u_list の情報から全ユーザの x と全ての $1 \leqq k \leqq \Delta$ に対して $g_1^{\frac{1}{x+k+T}}$ を計算する．τ と同じ計算結果が見つかったときはその x に対応した ID を，見つからなかったときは 0 を出力する．

4.4 安全性

以下，4.2 節で示したアルゴリズムの安全性について，以下のような観点から達成できている．なお，先行研究 [7] より議論されている匿名性，リンク不能性，追跡可能性に加え，Bloom filter の特性である偽陽性が発生する場合も考慮した．

(1) **Anonymity(匿名性), Unlinkability(リンク不能性)**
これは 2 つのユーザ ID のいずれかのユーザによる署名が与えられたとき，その署名がどちらの ID であるか推測できないという性質である．このとき，2 つの署名の署名者が同一かわからないこと（Unlinkability）も意味する．SDDHI 仮定より，失効タグ τ の値は疑似乱数と判別不可能である．よって第三者によりユーザの秘密情報 x，ならびに署名作成回数 k を識別することが不可能である．失効タグ以外は SPK であり，ゼロ知識性から第三者によるユーザの識別は不可能である．ここで k も秘匿されているため，k を用いた特定もできない．

(2) **Traceability（追跡可能性）**
これは結託したユーザ以外の ID もしくは失効されたユーザの ID に **Open** で追跡される署名を偽造することができない性質である．すなわち失効されたユーザの署名は検証に失敗することを意味する．署名 A_k をゼロ知識証明しており，その偽造不能性から，敵は honest なユーザとしてのグループ署名を作れない．また $1 \leqq k \leqq \Delta$ が保証されるため，不正な失効タグも作れないことから，失効ユーザの正当なグループ署名も作れない．

(3) **False Positive(偽陽性)**
Bloom filter の特性上，ユーザの認証時に偽陽性が発生することがある．しかしながら，ここでの偽陽性は「失効されていない正規のユーザ」が「失効されている」と示されることであり，失効されたユーザが不正に認証を通過することはない．こうして，偽陽性の発生によって，追跡可能性に影響を及ぼすことはない．

5. 評価

今回の評価では，単位期間あたりに生成できる署名の上限数 Δ を 100 に固定し，先行研究の方式の失効リストサイズと，提案方式の Bloom filter のデータサイズを，失効ユーザ 1 万人から 10 万人まで変化させながら比較した．なお，本研究は先行研究で使用している楕円曲線暗号ライブラリである ELiPS[8] を使用している．実装されている楕円曲線 BLS-12-461 に対して，$\mathbb{G}_1$ の元を 512bit で表現

するものとしており，τ のサイズも 512bit となる．

5.1 計測結果と評価

5.1.1 偽陽性確率を固定した際の失効者数に対するデータサイズの変化

まず，図 2 に偽陽性確率を 0.1%, 単位期間中における署名作成上限数 Δ を 100 に固定しながら失効者数を変化させた時の，元の失効リストと Bloom filter のデータサイズの変化を示す．計測の結果，両手法ともに失効者数に比例してデータサイズは増加するが，失効者数の大小に関係なく先行手法と比較し，約 96.3%減少という結果が得られた．Bloom filter が 1 つの要素あたりに必要とするデータサイズは $\frac{m}{n} = -1.44 \log_2 10^{-4} \approx 19.1$bit である．先行研究の方式では失効タグは 512bit であるため，96.3% の減少となる．

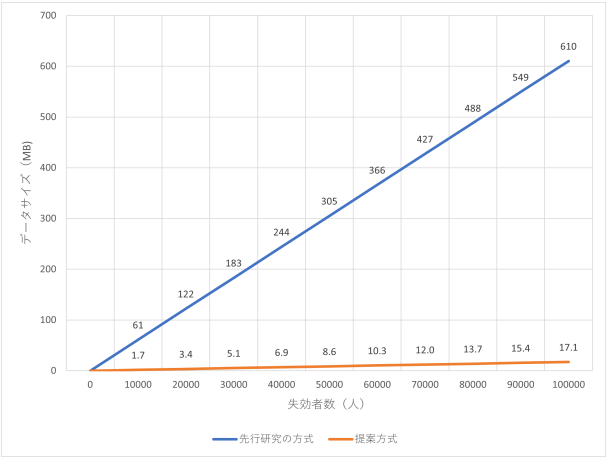

図 2 失効者数に対する失効リストのデータサイズの変化

5.1.2 失効者数を固定した際の偽陽性確率に対するデータサイズの変化

次に，図 3 に失効者数を 10,000 人，単位期間中における署名作成上限数 Δ を 100 に固定しながら, 偽陽性確率を変化させた時の Bloom filter のデータサイズの変化を示す．図 3 に示される通り，偽陽性確率が高確率であるほどデータサイズは減少する．これは，偽陽性確率が上昇するにつれ，Bloom filter が 1 要素あたりに必要とするデータ量が減少するためである．図 3 に示すように，偽陽性確率が 10^{-7} という低確率の条件下においても, 約 4MB となっている．同条件の場合，先行手法の失効リストサイズは約 61MB であるため，約 93 %低減という結果となった．なお，偽陽性の確率を 0.1 %とした場合，約 97 %の低減となる．実際に同手法を失効可能グループ署名による通信に適用する場合，それぞれの運用形態，ならびに許容される偽陽性の発生確率等を考慮し最適な Bloom filter を運用する必要がある．また，図 4 には，偽陽性確率毎の Bloom filter のデータサイズの変化を示す．図 2 と同様に，失効者数の増加に比例して Bloom filter を用いた失効リストのデータサイズは増加し，偽陽性確率が下がるに従い，その傾きが大きくなる．それでも，先行研究の方式の失効リストのデータサイズと比較し大幅にデータサイズが低減されている．

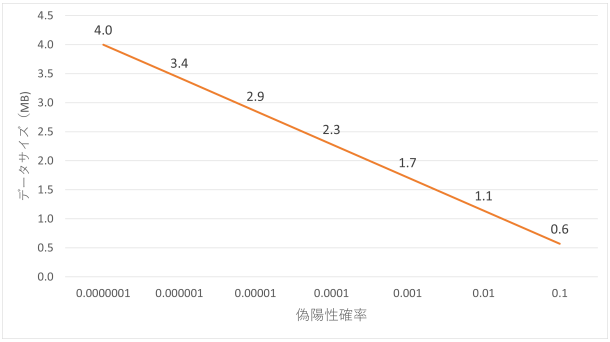

図 3 偽陽性確率に対するデータサイズの変化

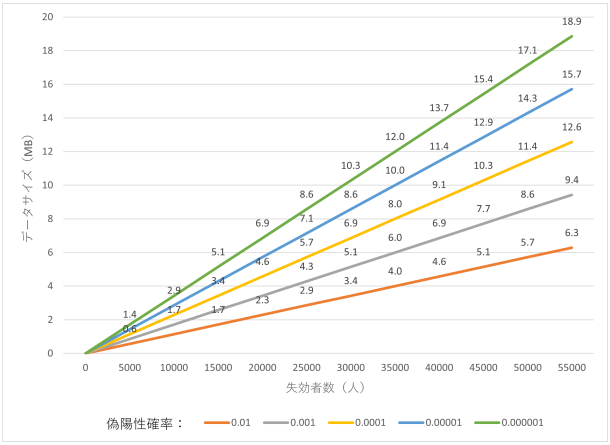

図 4 偽陽性確率毎の Bloom filter のデータサイズの変化

5.1.3 失効者数に対する Bloom filter の転送時間の比較

表 1 通信実験における PC のスペック

OS	Ubuntu 20.04.1 LTS
CPU	Intel® Core(IM) i5-9400 CPU @ 2.90GHz
メモリ	8.0GB

次に，GM から各検証者へ失効リストを配布することを想定し，失効者数を変化させながら先行研究での失効リストと，提案方式での Bloom filter の通信に要する時間を計測した．この通信実験では，表 1 の PC2 台を使用し，有線で学内ネットワークに接続して, Python を用いた通信プログラムによりデータ転送を行った．図 5 にその結果を示す．この図 5 が示すように，先行研究の方式に比べ，大幅に通信時間が低減している．失効者数が 7500 の場合，先行手法の失効リストのサイズは 48MB となり，通信時間は 4.04 秒要した．一方，提案手法では，失効リストのサイズは 1.8MB であり，通信時間は 0.15 秒となっていることから十分に高速化されている．これは失効者数だけでなく，

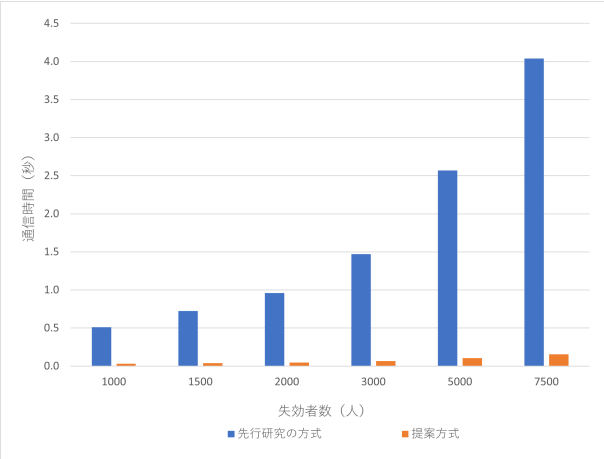

図 5 失効リストの通信時間の比較

単位期間内に生成できる署名の上限数 Δ の増加に対しても，失効リスト転送時間の低減を意味する．

6. まとめ

本研究では，従来方式の問題点である失効リストのデータサイズを Bloom filter を用いて削減する手法を提案した．また，失効者数，単位期間中に生成できる署名の生成上限数 Δ に比例して増加する失効タグ τ の数，また偽陽性の発生確率等を基に Bloom filter のデータを計測し，従来方式と比較した．その結果，偽陽性確率 0.1%の場合において，失効リストのデータサイズを約 3.7%にまで削減することが確認できた．また，今後の課題として，正規ユーザが認証に失敗した際の議論，より複雑な通信環境におけるシステムの検討と評価を行っていくことが挙げられる．

参考文献

[1] Dan Boneh, Xavier Boyen and Hovav Shacham, ”Short Group Signatures”, CRYPTO, pp.41-55, 2004.
[2] David Chaum and Eug ene van Heyst, ”Group Signatures”, EUROCRYPT ’91, pp.257-265, 1991.
[3] Raghu K. Ganti, Fan Ye, and Hui Lei, “Mobile Crowdsensing:Current State and FutureChallenges”, IEEE Communications Magazine Vol.49, No.11, pp.32-39, 2011.
[4] Sazzadur Rahaman, Long Cheng, Danfeng (Daphne) Yao, He Li, and Jung-Min(Jerry)Park, ”Provably Secure Anonymous-yet-Accountable Crowdsensing with Scalable Sublin-ear Revocation” Proceedings on Privacy Enhancing Technologies Vol.4, pp.287-306, 2017.
[5] Keita Emura and Takuya Hayashi, ” Road-to-Vehicle Communications With Time-Dependent Anonymity: A Lightweight Construction and Its Experimental Results”, IEEE Trans. Vehicular Technology, Vol.67, No.2, pp.1582-1597, 2018.
[6] Victor Sucasas, Georgios Mantas, Joaquim Bastos, Francisco Dami ao and Jonathan Ro-driguez, ”A Signature Scheme with Unlinkable-yet-Accountable Pseudonymity for Privacy-Preserving Crowdsensing”, IEEE Transactions on Mobile Computing, Vol. 19, No.4, pp.752-768, 2020.
[7] 中澤勇人, 中西透, 「クラウドセンシング向け失効可能グループ署名の高速化」, 信学技報, Vol.120, No.224, ISEC2020-34, pp.13-18, 2020.
[8] Yuto Takahashi, Yuki Nanjo, Takuya Kusaka, Yasuyuki Nogami, Tadaki Kanenari, Tomoya 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.