平成 30 年度

修士学位論文

検証可能な匿名情報収集方式を用いた電子投票の実現と応用

An E-voting System using Anonymous Data Secure Collection methods

1215098　　安光 穂高

指導教員　　清水 明宏

2019 年 2 月 27 日

高知工科大学大学院 工学研究科 基盤工学専攻

情報学コース

要 旨

検証可能な匿名情報収集方式を用いた
電子投票の実現と応用

安光 穂高

既存方式として，公開鍵暗号をベースとした手法がいくつか提案されているが，要件を満たしている方式においても，投票内容が制限されている方式や実現するための構築・運用コストが高い方式，投票者の計算負荷が高い方式など要件とは別の問題が存在する．

携帯端末等の処理能力の低い計算機での投票を想定し，投票内容の制限を無くした比較的シンプルな方式として匿名情報収集方式がある．この方式は，投票者の送信するデータを二つに分け，投票と秘密鍵を別々に管理させることにより匿名性を維持しつつ投票を集計できる投票プロトコルである．

しかし，匿名情報収集方式では，全てのエンティティが手順通りに正しくプロトコルを実行し，単独での不正を行わないという前提条件がある．実際に運用する上では，不正者がいた場合でも正しく実行されるプロトコルが望ましい．そこで，本提案では，集計者が不正を行う可能性があることを前提とし，集計結果の改ざんがあった場合でも第三者が検証を行うことで，不正を検知できる仕組みを実現する．

キーワード　ワンタイムパスワード，SAS，匿名情報収集，電子投票，匿名通報

Abstract

An E-voting System using Anonymous Data Secure Collection methods

Hodaka YASUMITSU

Several methods based on public key cryptography have been proposed as existing methods. In these methods, the calculation load of the voter is high. There is an anonymous information gathering method that allows aggregation of votes while maintaining anonymity by allowing voters and secret keys to be managed separately while taking account of the calculation load of voters. It is desirable to have a protocol that is executed correctly even if there is a fraudster. Therefore, in this proposal, on the premise that there is a possibility that the counterparty will carry out wrongdoing, even if there is tampering of the aggregation result, realize a mechanism capable of detecting falsification by verification.

key words one time password, SAS, collection of anonymous information, electronic voting, anonymous report

目次

図目次

表目次

第 1 章

序論

1.1 研究背景

インターネットの普及，パソコンやスマートフォン等の情報端末が身近な存在となってきたことにより，様々な情報がそれぞれの端末から発信され，インターネットを通してやり取りされている．こうした情報社会の発展に伴い，オンラインショッピングや店舗の予約など現実で行われてきた作業を効率よく，低コストで実現したサービスが増加している．そうした中，選挙において集計時のコスト削減や投票者の利便性向上による投票率アップを目的に，インターネット上で投票を行う電子投票に注目が集まっている．総務省では，インターネット上での投票を三段階に分けた電子投票の全国的な普及計画を立てている [1].

第一段階

選挙人が指定された投票所において電子投票機を用いて投票する段階

第二段階

指定された投票所以外の投票所においても投票できる段階

第三段階

投票所での投票を義務付けず，個人の所有する情報端末を用いて投票する段階

この第三段階における電子投票は，投票者の匿名性保護や集計結果の改ざん等のセキュリティ上の問題や費用対効果の面から実現は難しいとされている．電子投票における既存方式では，準同型暗号を利用した方式 [2][3] やブラインド署名を利用した方式 [4][5]，匿名通信路を実現している Mixnet を利用した方式 [6][7][8]，ブロックチェーンを利用した方式 [9] な

どが提案されている．しかし，これらの既存方式には投票内容に強い制約があるものやシステムを実現するための構築・運用コストが高いもの，集計に時間がかかるなどの問題が存在する．

1.2　電子投票方式が満たすべき要件

電子投票方式が満たすべき条件 [10] を以下に示す．

Availability 可用性

投票システムは，選挙期間中はずっと利用可能である必要がある．

Eligibility 適格性

有権者だけが 1 つの有効な投票を行うことができる．二重投票は許されない．

Integrity 完全性

投票の完全性が保証されなければならない．

Anonymity 匿名性

投票者と投票の対応付けは，投票者自身が漏洩させない限り秘密である．

Fairness 公平性

集計が終了する前に部分的あるいは全ての集計結果が公開されない．

Receipt Freeness 無証拠性

強制や買収を減らすために，投票者は自分の投票についての情報を得られない．つまり，投票者は自分の投票を証明できない．

Correctness 正しさ

選挙結果は正しく数えられ，正しく公表される．

Robustness 堅牢性

何らかの誤りが混入した場合でも，それを排除できる．

Universal Verifiability 第三者検証可能性

集計結果は，誰でも検証することができる．

Voter Verifiable 投票者検証可能性

投票者自身が自分の投票が適切にカウントされたことを確認できる．

Coercion 強制

強制者の存在下でも，投票者の意図通りに投票できる．これは，売却されたまたは漏洩した信任状でさえ使用できないようにすることを意味する．

1.3　本論文の構成

本論文は，全 7 章で構成される．本章では，研究背景および電子投票が満たすべき要件を述べた．第 2 章では，本研究で必要となる要素技術について簡単に説明する．第 3 章では，電子投票における既存方式について説明し，問題点を示す．第 4 章では，簡易で安全な投票プロトコルである匿名情報収集方式について述べ，集計者の不正を考慮した場合に集計結果の正真性を検証できないことを示す．第 5 章では，匿名情報収集方式において検証可能な投票プロトコルを提案する．第 6 章では，提案方式が検証可能であることを示し，増加した計算負荷について評価を行う．また，評価結果を踏まえ適用可能な投票規模と運用上の注意について検討する．第 7 章では，本論文のまとめを行い，今後の課題を述べる．

第 2 章

要素技術

2.1　暗号技術

通信上のセキュリティを保つため，暗号技術は必要不可欠である．以下では，暗号技術の基礎である，共通鍵暗号，公開鍵暗号および一方向ハッシュ関数について述べる [11][12].

2.1.1　共通鍵暗号

共通鍵暗号は，暗号化と復号に同じ鍵を用いる暗号である．共通鍵暗号では送信者が受信者に秘密のメッセージを送りたい場合，送信者と受信者で同じ鍵を安全に共有しておく必要がある．

2.1.2　公開鍵暗号

公開鍵暗号は，暗号化する鍵と復号する鍵が異なる暗号である．公開鍵暗号の鍵は対になっており，公開して使用する公開鍵と自分だけが使用する秘密鍵がある．公開鍵は自分以外に知られてもよいため，受信者が公開鍵を公開していれば，送信者はその公開鍵を使って暗号化し，受信者に送ればよい．公開鍵で暗号化されたメッセージを復号できるのは，秘密鍵を持つ受信者のみである．つまり，公開鍵暗号では，事前に鍵を共有しておく必要がない．しかし，公開鍵暗号の暗号化・復号処理は共通鍵暗号に比べ複雑になっているため，一般に処理時間が数百〜数千倍かかる．また，通信相手が正しいことを証明する，電子署名に応用できる．電子署名として利用する際は，自身の秘密鍵で暗号化することで実現できる．

2.1.3 一方向ハッシュ関数

一方向ハッシュ関数は，あるメッセージを入力として受け取り，固定長のハッシュ値を出力する関数である．SHA-256 という一方向ハッシュ関数では，ハッシュ値は常に 256 ビット (32 バイト) となる．メッセージのデータサイズに関係なく固定長のハッシュ値を出力し，メッセージが 1 ビットでも異なれば出力するハッシュ値は全く異なるため，改ざん検知などに用いられる．以降，一方向ハッシュ関数を「ハッシュ関数」と表記する．

2.2 ブロックチェーン

ブロックチェーンは，分散型台帳として各分野での応用が期待される汎用性の高い技術であり，いくつかの暗号技術の組み合わせにより実現される．電子署名とハッシュ関数を利用して，偽造や改ざんの防止を可能としている．ブロックチェーンでは，取引記録を「ブロック」という単位でまとめ，時系列で連なっているため，過去のブロック内容を改ざんすることは困難である．また，分散型台帳を実現するため，ネットワークに参加する全ての人が取引記録を確認できる．つまり，取引の透明性が高いことも特徴の 1 つである．

2.3 SAS ワンタイムパスワード認証方式

この節では，共通鍵暗号方式の応用であるワンタイムパスワード認証方式 SAS-2[14] について説明する．簡単化のためにリプレイアタック [15] 対策の仕様を外して説明する．

SAS-2 は，登録と認証の 2 つのフェーズから構成される．登録フェーズは初回に一度のみ行い，認証フェーズは認証が必要となる度に行う．図 2.1 に SAS-2 の登録フェーズを示す．ユーザはユーザ識別情報 ID と秘密に保持しているパスワード情報 S を入力し，さらに乱数 R_0 を生成する．ハッシュ関数を H，排他的論理和を $\oplus$ として，初回の認証で用いる認証情報 $A_0 = H(R_0 \oplus S)$ を算出し，ID とともに安全なルートでサーバへ送信する．サーバは受け取った ID と A_0 を登録する．図 2.2 に SAS-2 の $i+1$ 回目の認証フェーズを示す．ユーザは，ID とパスワード S，R_i および新しく生成した乱数 R_{i+1} を用いて，

$\alpha = H(H(R_{i+1} \oplus S)) \oplus H(R_i \oplus S)$ と $\beta = H(R_{i+1} \oplus S) \oplus H(R_i \oplus S)$ を算出し，認証サーバへ送信する．サーバは，登録している認証情報 $A_i \oplus \alpha$ から $H(H(R_{i+1} \oplus S))$ を算出し，$H(R_i \oplus S) \oplus \beta$ で得られる $H(R_{i+1} \oplus S)$ にもう一度ハッシュ関数を適用して得られる情報と同じであれば，正しいユーザであると認証する．$H(R_{i+1} \oplus S)$ を次回認証情報として登録する．$H(R_{i+1} \oplus S)$ に再度ハッシュ関数を適用した $H(H(R_{i+1} \oplus S))$ をユーザに送信する．ユーザは，サーバから受け取った $H(H(R_{i+1} \oplus S))$ と A_{i+1} にハッシュ関数を適用したものが同じであるかを確認し，同じであればサーバの認証の完了を確認する [14].

以上説明した SAS-2 は，毎回ハッシュ関数 3 回の適用で認証を実現する安全かつ高速なワンタイムパスワード認証方式である．ハッシュ関数は共通鍵暗号方式を用いて実現できる．

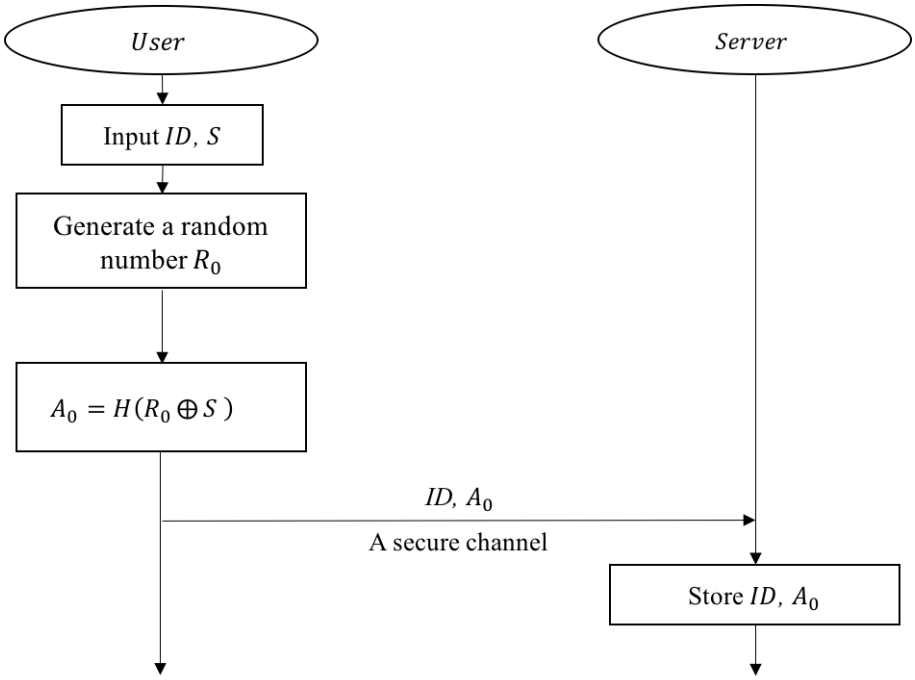

図 2.1　SAS-2 の登録フェーズ

2.3　SAS ワンタイムパスワード認証方式

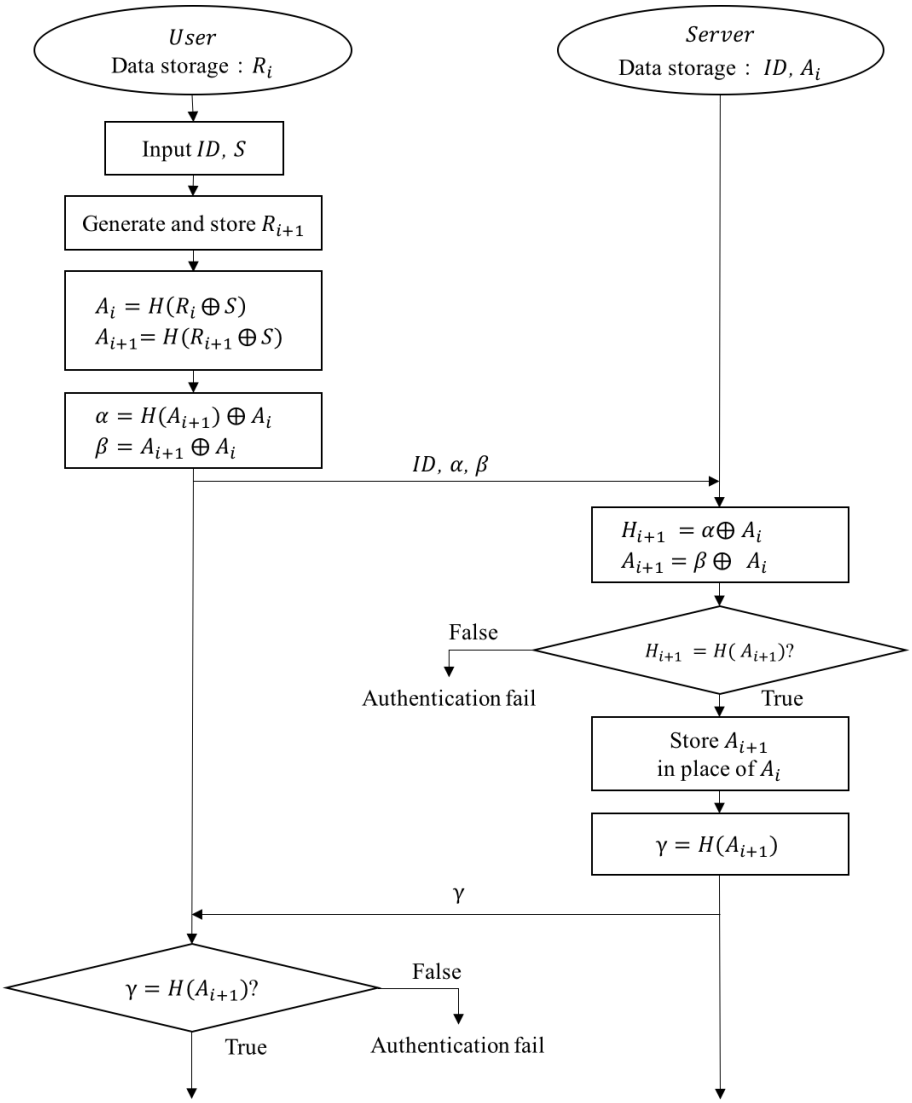

図 2.2　SAS-2 の認証フェーズ

第 3 章

公開鍵暗号ベースの既存方式

3.1　準同型暗号利用方式

準同型暗号利用方式は，投票者と集計者で構成され，投票内容が二値であることから，信任投票 (賛成か反対か) を基本に構成される方式である．投票内容が 0 か 1 かであることと，準同型暗号の性質を利用し，暗号化した状態で集計することが可能である．そのため，集計者が投票内容を知ること無く集計でき，一度復号するだけで集計結果を得られる利点がある．ただし，候補者が複数いる場合には，その数だけ並行処理が必要となり，処理負荷と通信量が大きく増加する．そのため，大規模な選挙には向いていない．また，投票内容が二値であるため，自由記述などの回答はできないという問題がある [2][3].

3.2　ブラインド署名利用方式

ブラインド署名利用方式は，投票者と選挙管理人，集計者で構成される．この方式は，投票者が自身の投票に選挙管理人の電子署名を付けてもらい，匿名通信路を用いて集計者に送る．集計者は，選挙管理人の署名が付いているかを確認することで，正しい投票者による投票だということがわかる．匿名通信路とは，送信されたデータから送信者が特定できない通信路のことを指す．インターネットなどの通信路では通常匿名性は保証されていない．匿名通信路の構築には，複数のサーバを用意するなど，多大な構築コストがかかるという問題がある [4][5].

3.3 Mixnet 利用方式

Mixnet 利用方式は，匿名通信路を実現する Mixnet を利用した方式である．Mixnet 利用方式は，投票者と複数のサーバから構成されており，投票者は各サーバが公開している公開鍵を用いて，自身の投票を多重に暗号化する．多重に暗号化した投票に署名を付け，サーバに送る．受け取ったサーバは，署名から正しい投票者であることを確認し，秘密鍵で復号したのちシャッフルして次のサーバに送信する．この手順を繰り返すことで，匿名性を保ったまま投票できる．しかし，複数のサーバ全てが復号およびシャッフルを行うため，これらの復号処理に時間がかかるという問題がある [6][7][8].

3.4 ブロックチェーン利用方式

ブロックチェーンを用いて，投票権の付与・投票権の授受・投票の集計を行う取り組みが始まっている．ブロックチェーンのプラットフォームから発行されるトークンを投票権として利用することで，誰が有権者であるのかを電子的な管理で実現でき，投票権譲渡の記録管理や得票集計と言った業務を効率化できる．ただし，ブロックチェーンを利用した投票を行う場合，投票者がブロックチェーンのプラットフォームにアクセスできる環境を保持している必要があるため，大規模投票での活用は難しい．また，誰が誰に投票したのかを秘匿する必要がある投票においては，秘匿性を担保する仕組みを併せて構築する必要がある [9].

3.5 公開鍵暗号ベースの電子投票方式における問題点

以上のように，公開鍵暗号ベースの既存方式において，要件以外にも実用上の問題がある．特に第三段階での電子投票を考えると，投票者の計算負荷を高い場合，携帯端末等の処理能力の低い計算機では投票自体が難しい．そこで，学部での研究成果として，処理能力の低い携帯端末での利用を考慮した投票プロトコルである匿名情報収集方式 [13] を提案した．

第 4 章

匿名情報収集方式

4.1　システムの概要

この方式 [13] は，投票者，投票を保管する保管サーバ，暗号鍵を保管する鍵サーバ，投票を集計する集計者から構成される．投票者は投票を暗号化し，保管サーバに送る．また，投票者は投票の暗号化に用いた暗号鍵を暗号化し，鍵サーバに送る．このように投票と暗号鍵の管理を分けることで，集計者は匿名性を維持したまま安全に投票を収集することができる．この方式の概要図を図 4.1 に示す．処理手順は，登録フェーズ，鍵準備フェーズ，投票フェーズ，集計フェーズ，公表フェーズの 5 つからなる．

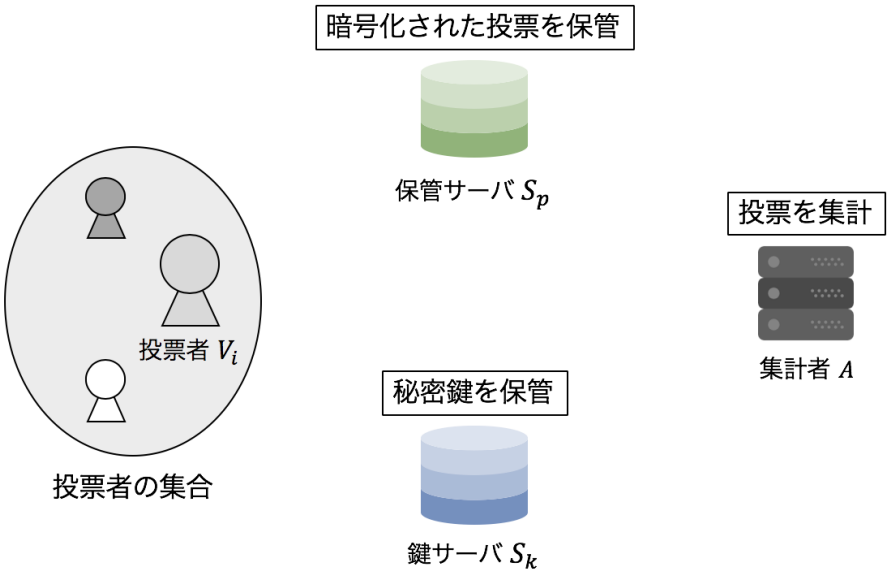

図 4.1　匿名情報収集方式の概要図

4.2　定義と記法

表 4.1 に匿名情報収集方式で用いる記号の定義と記法について記述する．

表 4.1　提案方式における定義と記法

記号	定義
V_i	i 番目の投票者．ただし，$(1 \leq i \leq n)$
AD	投票を集計する集計者．
SP	投票を保管する保管サーバ．
SK	秘密鍵の保管と集計結果の検証を行う鍵サーバ．
Cid_j	j 番目の候補者を示す ID．ただし，$(1 \leq j \leq m)$
K_G	AD と投票者全員のグループ鍵．
P_i	V_i の秘密鍵．V_i 自身が生成する．
A_i	V_i と AD で共通の暗号鍵．投票者の認証情報を元に生成される．
$H(x)$	データ x をハッシュ関数 H によって変換した値．
$Enc(x,k)$	データ x を暗号鍵 k で暗号化した値．

これ以降，特別な説明が無い限り，添字に i が付くものは i 番目の投票者が所持するものを指すこととする．また，投票者は n 人，候補者は m 人存在することとして説明を進める．

4.3　登録フェーズ

登録フェーズでは，投票者が各サーバと事前に認証情報を共有する．初めて投票を行う投票者は各サーバと初回認証情報を共有し，図 4.2 のように，それぞれのサーバと SAS 相互認証を確立する．SAS 認証方式では，初回認証情報の共有方法が議論の対象となる．匿名情報収集方式では実際の選挙での運用を想定しており，市役所等で発行したハガキを利用して，初回のみ市役所等の窓口で本人認証を行い，初回認証情報登録の手続きを行う．

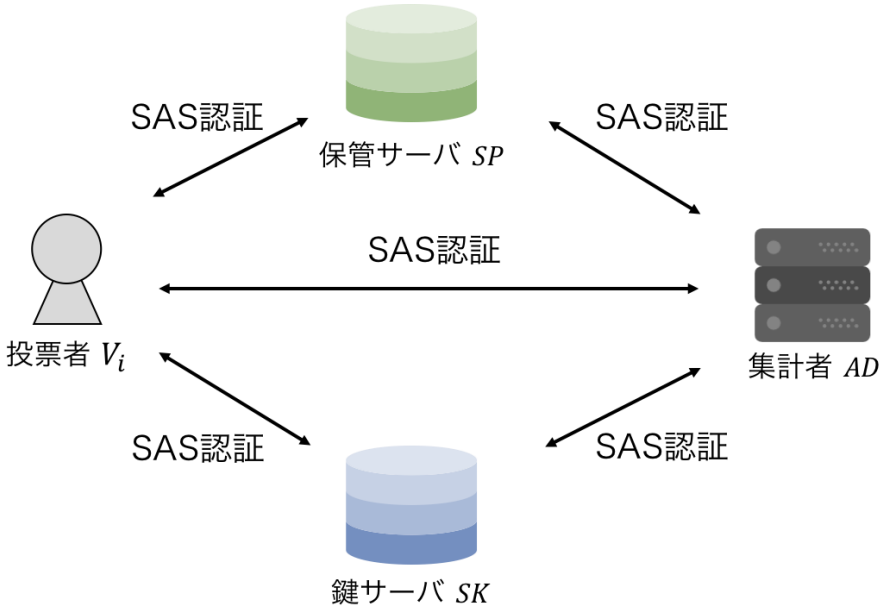

図 4.2　登録フェーズ

4.4　鍵準備フェーズ

鍵準備フェーズでは，図 4.3 に示すように，投票者および集計者が使用する暗号鍵を準備する．まず，集計者はそれぞれの投票者との認証情報をシードに共通鍵 $\{A_1, A_2, ..., A_n\}$ を生成する．投票者 V_i も同様に認証情報をシーズに共通鍵 A_i を生成する．

次に，集計者はグループ鍵 K_G を生成し，投票者に配布する．図 4.4 のように，集計者はグループ鍵 K_G を各投票者との共通鍵で暗号化して掲示板等に公開し，投票者が掲示板で公開されている鍵リストを復号することで，有権者のみがグループ鍵 K_G を入手することがで

きる．

最後に，各投票者は秘密鍵 P_i を生成する．秘密鍵 P_i は各投票者が自分自身で生成するため，投票者自身が漏らさない限り，他の参加者は秘密鍵 P_i に関する情報を知ることができない．

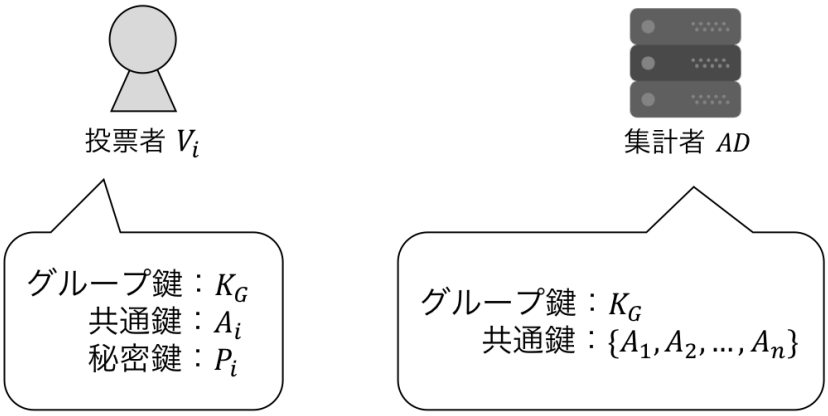

図 4.3　投票者と集計者の所持する暗号鍵

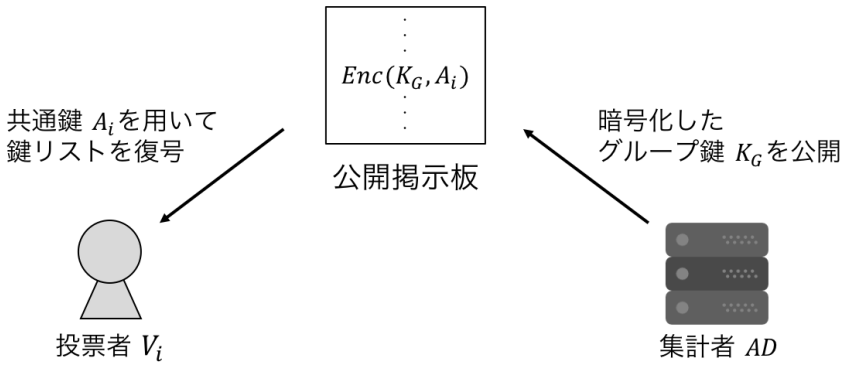

図 4.4　掲示板を用いた鍵共有

4.5　投票フェーズ

投票フェーズでは，図 4.5 のように，投票者 V_i が保管サーバ SP に投票 v_i を送信し，投票の復号に必要な投票鍵 vk_i を鍵サーバ SK に送信する．以下に詳細を示す．

1. 投票者 V_i は，投票する候補者 ID のリスト $\{Cid_1, Cid_2, ..., Cid_m\}$ から一つ選び，その候補者 ID を秘密鍵 P_i で暗号化した後グループ鍵 K_G で暗号化し，投票 v_i を生成する．

   $v_i = Enc(Enc(Cid_j, P_i), K_G)$
2. 生成した投票 v_i を保管サーバ SP に送る．
3. 投票者 V_i は，秘密鍵 P_i をグループ鍵 K_G で暗号化した後共通鍵 A_i で暗号化し，投票鍵 vk_i を生成する．

   $vk_i = Enc(Enc(P_i, K_G), A_i)$
4. 生成した投票鍵 vk_i を鍵サーバ SK に送る．

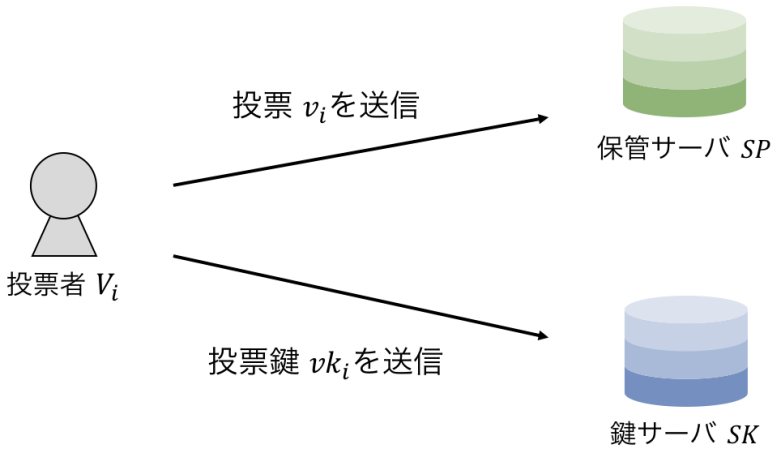

図 4.5　投票フェーズの概要図

4.6　集計フェーズ

集計フェーズでは，図 4.6 のように，保管サーバ SP は投票を集計者に送信し，鍵サーバ SK は投票鍵を集計者に送信する．保管サーバ SP は，投票 $\{v_1, v_2, ..., v_n\}$ をランダムに並

び替えて集計者に送る．鍵サーバ SK が投票者から受信した情報は，各投票者と集計者の共通鍵で暗号化されているため，何も処理しないで集計者に送ると，集計者は秘密鍵がどの投票者のものか紐づけすることが可能となってしまう．すなわち，鍵サーバが所持している間に，集計者が投票者を特定できないようにする必要がある．以下に詳細を示す．

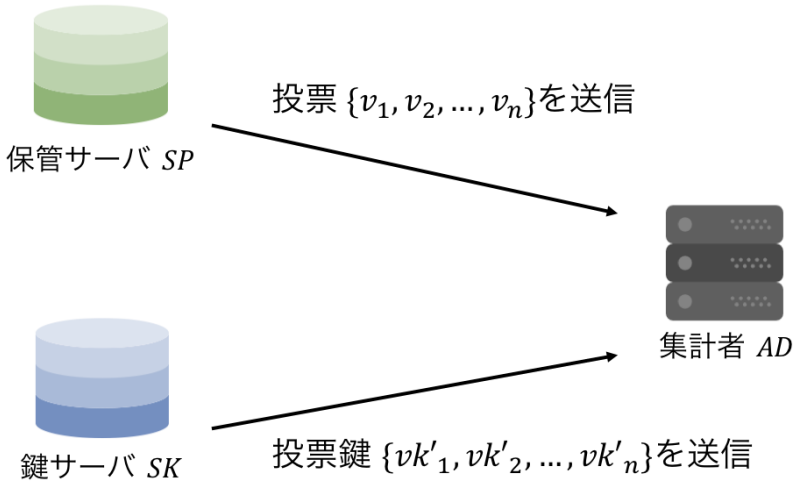

図 4.6　集計フェーズの概要図

1. 集計者は，各投票者との共通鍵 $\{A_1, A_2, ..., A_n\}$ を鍵サーバ SK に送る．
2. 鍵サーバ SK は，集計者から送られてきた鍵リストを用いて，投票者から受け取った投票鍵 $\{vk_1, vk_2, ..., vk_n\}$ を総当たりで復号し，vk_i' を生成する．

   $vk_i' = Enc(P_i, K_G)$
3. $\{vk_1', vk_2', ..., vk_n'\}$ を集計者に送る．
4. 集計者は，鍵サーバ SK から受け取ったデータを全ての投票者とのグループ鍵 K_G を用いて復号し，各投票者の秘密鍵 $\{P_1, P_2, ..., P_n\}$ を入手する．このとき，集計者はどの秘密鍵がどの投票者のものであるか特定できない．なぜなら，秘密鍵 $\{P_1, P_2, ..., P_n\}$ は各投票者が自分自身で生成したものであり，投票者自身が生成した秘密鍵を公開していない限り投票者以外に知られることは無いからである．
5. 最後に，集計者はグループ鍵 K_G および秘密鍵 $\{P_1, P_2, ..., P_n\}$ を用いて，保管サーバ

SP から受け取った投票 $\{v_1, v_2, ..., v_n\}$ を復号し，候補者 ID 毎の得票数を集計する．

4.7　公表フェーズ

公表フェーズでは，図 4.7 のように，集計者が「集計に使用した秘密鍵のハッシュ値」を公開する．ここで使用するハッシュ関数はアルゴリズムが公開されており，事前に集計者と投票者間で共有されているとする．使用するハッシュ関数の具体例には，SHA-256 などが挙げられる．

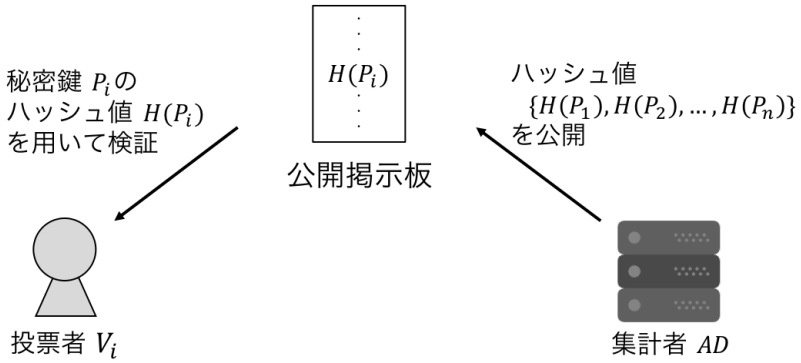

図 4.7　公表フェーズの概要図

1. 集計者は，投票の復号に用いた秘密鍵 $\{P_1, P_2, ..., P_n\}$ にハッシュ関数を適用し，掲示板等を用いて，投票者に公開する．
2. 投票者 V_i は，自身の秘密鍵 P_i にハッシュ関数を適用し，集計者が公開したものと一致するかを確認する．一致するデータがあれば，自身の秘密鍵が集計者に正しく到達していることがわかり，集計者が正しくプロトコルを実行している限り，集計結果に自身の投票が反映されていることが検証できる．
3. 集計者は，投票者からの異議申し立てが無ければ，掲示板等を用いて集計結果を公開する．

4.8　既存方式の問題点

既存の匿名情報収集方式では，集計者が復号した投票を集計し公表するため，集計時に不正の余地がある．投票者は公表された結果に集計者の不正が存在しないことを検証したいが，既存方式で提案されている検証方法では，集計者による集計結果の改ざんは検知できない．

4.8.1　集計者による集計結果の改ざん

図 4.8 に想定している集計結果を示す．候補者 ID Cid_j の得票数を $t_j (0 \leq t_j \leq n)$ とする．

候補者ID	Cid_1	Cid_2	…	Cid_m
得票数	t_1	t_2	…	t_m

$$t_1 + t_2 + \cdots + t_m = n$$

n は投票者の人数，m は候補者の人数を表す．

図 4.8　集計結果

集計者が集計結果を改ざんする場合，図 4.9 のように得票数を操作することが挙げられる．得票数の操作として，水増しや削除，入れ替えなどがある．

既存の検証方法では，集計者が検証データとして公開するのは，復号に用いた各投票者の秘密鍵のハッシュ値 $\{H(P_1), H(P_2), ..., H(P_n)\}$ であるため，投票者が検証できるのは「自身の秘密鍵が集計者に届いたかどうか」だけである．

集計結果の正当性を検証するには，単純に考えると，投票者または第三者が集計者と同様に投票者全員の投票を集計し検証することが必要となる．しかし，投票者が集計者と同様の処理をするとなると，投票者の負担が大きくなり，投票システムとしての利便性を大きく損なうことになる．

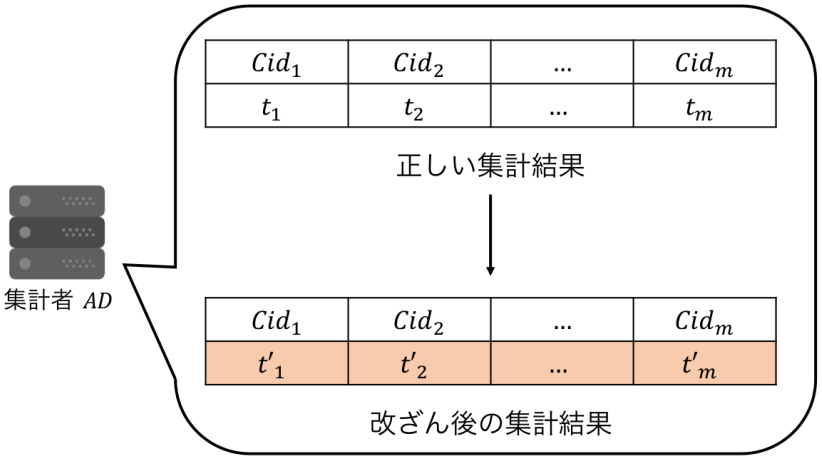

図 4.9　集計結果の改ざん

第 5 章

提案方式

本提案方式では，投票者の負担を抑えつつ，匿名情報収集方式における集計結果を検証できる手法を示す．

5.1 前提条件

集計者は，集計結果の改ざんを行う可能性があり，その他のエンティティは決められた手順通りにプロトコルを実行するものとする．

5.2 検証方法の概要

匿名情報収集方式において，集計結果の検証は鍵サーバ SK が行う．投票者が検証データを作成し，鍵サーバは，集計者が集計した結果と投票者が作成した検証データを用いて集計結果が等しくなることを検証する．図 5.1 に検証の流れを示す．

1. 投票者 V_i は，秘密鍵 P_i にハッシュ関数を適用し，$H(P_i)$ を生成する．
2. ハッシュ化した秘密鍵 $H(P_i)$ で投票 v_i を暗号化し，検証データ $f_i = Enc(v_i, H(P_i))$ を生成する．
3. 検証データ f_i を鍵サーバ SK に送る．
4. 投票者 V_i は，秘密鍵 P_i をグループ鍵 K_G で暗号化した後共通鍵 A_i で暗号化し，投票鍵 vk_i を生成する．

   $vk_i = Enc(Enc(P_i, K_G), A_i)$

5. 生成した投票鍵 vk_i を鍵サーバ SK に送る．
6. 保管サーバ SP は，投票者から受け取った検証データ $f_1, f_2, ..., f_n$ を鍵サーバ SK に送る．このとき，それぞれの検証データがどの投票者から受け取ったものであるかは伝えない．
7. 集計者は，投票の復号に用いた秘密鍵 $P_1, P_2, ..., P_n$ に対して，ハッシュ関数を適用し，検証鍵 $fk_1, fk_2, ..., fk_n$ を生成する．
   $fk_i = H(P_i)$ である．
8. 集計者は，集計結果と検証鍵 $fk_1, fk_2, ..., fk_n$，候補者 ID のリスト $\{Cid_1, Cid_2, ..., Cid_m\}$ を鍵サーバ SK に送る．
9. 鍵サーバ SK は，検証鍵 $fk_1, fk_2, ..., fk_n$ で候補者 ID のリスト $\{Cid_1, Cid_2, ..., Cid_m\}$ から一つずつ暗号化し，検証データ $f_1, f_2, ..., f_n$ と比較して一致するものがあるかどうかを検証する．一致するものの数と集計結果に示されている候補者 ID 毎の得票数を比較し，集計者の集計行為が正しいことを確認する．
10. 鍵サーバ SK が検証に成功した場合は，集計結果を公表する．検証に失敗した場合は，一致しなかった検証データを公開する．

図 5.2 のように，投票者から鍵サーバ SK へ検証データを送ると，投票者と検証データの対応がわかり，検証時にどの投票者がどの候補者に投票したのかがわかってしまう．そのため，投票者は，検証データを投票フェーズにて保管サーバ SP に送り，公表フェーズにて，保管サーバ SP から鍵サーバ SK へ検証データを送ることとする．無論，保管サーバは鍵サーバに検証データを送る際に，投票者との対応は秘密にしておく．

5.2　検証方法の概要

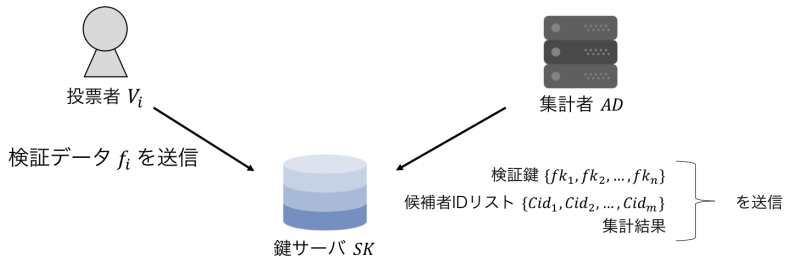

図 5.1　集計結果の検証

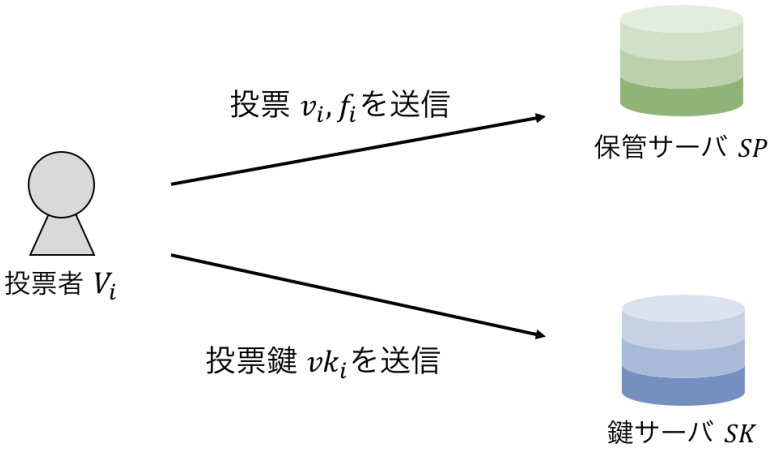

図 5.2　検証データの送信

第 6 章

評価

まず，本提案方式の根幹である，検証可能となったこと示す．また，その他の要件に関しても満たしているもの，満たしていないものを明確にする．その後，増加した処理手順による各エンティティの計算量を既存の匿名情報収集方式と比較する．投票規模について検討および，運用上注意することについて説明する．

6.1 集計の検証可能性

集計者が集計結果を改ざんした場合，検証によって改ざん検知できることを示す．集計者は，候補者 ID Cid_j の得票数を t'_j に改ざんしたとする．すなわち，$t_j \neq t'_j$ が成り立つ．

集計者は，検証鍵 $\{H(P_1), H(P_2), ..., H(P_n)\}$ を正しく鍵サーバに送ることとする．なぜなら，鍵の数が足りない場合は，鍵サーバが持つ検証データの数と一致しないため，不正が発覚する．また，偽の検証鍵 $H(P'_i)$ を生成した場合，鍵サーバが生成する検証データ f'_i と投票者が作成した検証データ f_i が一致せず，不正が発覚する．

したがって，集計者による得票数の改ざんを検知できる．

6.2 その他の要件

1.2 節に示した電子投票の要件を，集計者による不正を仮定した場合に満たしているかどうかを示す．また，それぞれのエンティティは結託を行わないものとする．

6.2.1　Availability 可用性

本提案システムでは，投票と暗号鍵の管理を二つのサーバに分けているため，どちらのサーバが停止した場合でも単一障害点となりうる．つまり，本提案システムのみでは，可用性を満たすことはできない．そのため，運用する上ではサーバ自体の冗長化などの対策が必須となる．

6.2.2　Eligibility 適格性

有権者の確認と二重投票の防止が満たせていれば，適格性を満たすことができる．

有権者の確認は，集計者が，グループ鍵 K_G を生成し共通鍵 A_i で暗号化して配布するため，集計者に認証されている投票者のみ，グループ鍵 K_G を入手することができる．仮に不正者がグループ鍵 K_G を入手して，不正な投票を行おうとしても，共通鍵 A_i がないため，集計フェーズ時に鍵サーバ SK が鍵リストを用いて復号する際に取り除くことができる．

したがって，有権者の確認ができている．

次に二重投票の防止を考える．不正な投票者が二回以上投票しようとした場合，以下のような不正が考えられる．

① 保管サーバ SP に複数の投票を送り，鍵サーバ SK に複数の暗号化された鍵情報を送る．

② 一つの秘密鍵 P_i で複数の投票を暗号化して，保管サーバ SP に複数の投票を送り，鍵サーバ SK には一つの秘密鍵 P_i のみ送る．

共通鍵 A_i は各投票者に一つであるため，有権者確認可能性と同様に，集計フェーズ時に鍵サーバ SK が鍵リストを用いて復号する際に取り除くことができ，集計者に無効な鍵が存在したことを報告できる．また，不正な投票者が投票を複数生成し，保管サーバ SP に送ろうとしても，ユーザ認証を行っているため，複数投票していることが保管サーバ SP にわかってしまう．

したがって，①，②どちらの状況でも二重投票を防ぐことができる．

以上から，適格性を満たすことができる．

6.2.3　Integrity 誠実さ

投票の完全性は，投票者がプロトコルに従って処理を行う限り保証されている．しかし，運用上は投票者の不正や攻撃者による不正を考慮する必要があるため，不正に対する耐性があることを検証する必要がある．

6.2.4　Anonymity 匿名性

匿名性の要件を満たしているかどうかは，投票や投票鍵の情報から，誰が候補者 ID Cid_j に投票したのかを特定できなければ満たしていると言える．投票者 V_i が各サーバに送る情報として，投票 v_i と投票鍵 vk_i がある．投票 $v_i = Enc(Enc(Cid_j, P_i), K_G)$ であり，投票鍵 $vk_i = Enc(Enc(P_i, K_G), A_i)$ である．Cid_j を入手可能なエンティティが，どの投票者から届いた投票か特定できるかが問題となる．

秘密鍵 P_i は投票者 V_i が生成したものであり，投票者 V_i 以外のエンティティは知らない．

保管サーバ SP は，投票 v_i を保持しているが，グループ鍵 K_G および，秘密鍵 P_i を知らないため，投票 v_i を復号できない．

鍵サーバ SK は，投票鍵 vk_i を保持しているが，投票者 V_i と集計者間の認証情報から生成された共通鍵 A_i およびグループ鍵 K_G を知らないため，秘密鍵 P_i を知ることができない．集計フェーズが実行されると，鍵サーバ SK は共通鍵 A_i を入手でき，復号することで $vk'_i = Enc(P_i, K_G)$ を得られるが，グループ鍵 K_G がわからないため，秘密鍵 P_i は特定できない．また，検証可能にしたことにより，候補者 ID のリスト $\{Cid_1, Cid_2, ..., Cid_m\}$ を入手するが，暗号化して検証データとの比較を行っても，どの投票者の検証データかを知らないため，投票者の匿名性は保たれる．

集計者が入手する情報は，投票 v_i と投票鍵 vk'_i であり，どちらもグループ鍵 K_G で復号

できるため，保管サーバや鍵サーバがそれらのデータと投票者との関連を教えなければ，集計者は得られた秘密鍵 P_i と投票 v_i から投票者を特定することができない．

したがって，匿名性の条件を満たすことができている．

6.2.5　Fairness 公平性

投票の復号は集計フェーズにて実行されるため，投票の前に投票結果の一部が知られることはない．

したがって，公平性の条件を満たすことができている．

6.2.6　Receipt Freeness 無証拠性

本提案で実現した集計結果の検証は，鍵サーバが行うため，投票者自身が集計結果を検証することはない．

したがって，投票者は自分自身が誰に投票したのかを投票後に証明することができないため，無証拠性を満たすことができている．

6.2.7　Correctness 正しさ

検証可能性が満たせているため，集計結果の検証を鍵サーバが正しく行えば，正しさを満たすことができる．

6.2.8　Robustness 堅牢性

前提条件として集計者の不正を考慮するため，集計者の不正によって生じる改ざん等の誤りを排除できることを示す集計者以外のエンティティがプロトコル通りに処理を実行すれば，集計者の不正は鍵サーバによる集計結果の検証から容易に検知できる．ただし，実用上は投票者，保管サーバおよび鍵サーバの不正も考慮する必要があるため，それぞれのエンティティの正当性を保証する仕組みを検討する必要がある．

したがって，集計者の不正のみを考慮した場合は堅牢性を満たすことができているが，今後実用を目指す上でその他のエンティティの不正についても検討する必要がある．

6.2.9　Universal Verifiability 第三者検証可能性

集計結果の検証は，鍵サーバのみが行い，公表された結果をその他の第三者が検証することはできない．そのため，第三者検証可能性は満たさない．

6.2.10　Voter Verifiable 投票者確認可能性

投票者は検証データを作成するが，検証自体は行わないため，投票者確認可能性は満たさない．

6.2.11　Coercion 強制

投票フェーズ時に，投票者が強制者に投票を強制される場合，提案システムでは防ぐことができない．運用上の対策を考慮すると，投票場所を限定することで，監視によりその場所での強制には耐性を持つが，事前に買収されている場合は対処できない．

6.3　計算量の増分比較

提案方式では，集計結果の検証を可能にするため，投票者および鍵サーバの計算量が増加している．表 6.1 に計算量を比較した結果を示す．比較項目は，ハッシュ関数の適用回数，暗号化処理回数，データの比較回数である．

投票者は既存方式に比べ，検証データ $f_i = Enc(Cid_j, H(P_i))$ を作成する処理が増加しているため，ハッシュ関数の適用回数と暗号化処理回数がそれぞれ 1 回ずつ増加している．暗号アルゴリズムに AES-128 を用いる場合，ハッシュ関数を適用する秘密鍵 P_i の鍵長は 128bit となり，処理負荷は少ない．また，候補者 ID Cid_j を暗号化するが，候補者 ID は候補者を識別できれば良いため，データサイズは小さくなる．よって，提案方式による投票者

の計算負荷は，実用上問題ない範囲に抑えられる．

鍵サーバは既存方式に比べ，暗号化処理回数が mn 回，データの比較回数が $\frac{m(n^2+n)+2n}{4}$ 回増加している．

表 6.1　計算量の比較（単位：回）

	既存方式			提案方式		
	ハッシュ関数	暗号化	比較	ハッシュ関数	暗号化	比較
投票者	1	4	1	1	5	0
鍵サーバ	0	0	0	0	mn	$\frac{m(n^2+n)+2n}{4}$

6.4　具体的な適用箇所の検討

具体的な適用箇所として，購入した商品に付属する投票権を用いた Web 上での投票を想定する．本方式では，投票権を入手した有権者と各サーバがどのように暗号鍵を共有するかを適用箇所ごとに考慮しなければならない．図 6.1 に想定モデルでの鍵共有方法について示す．

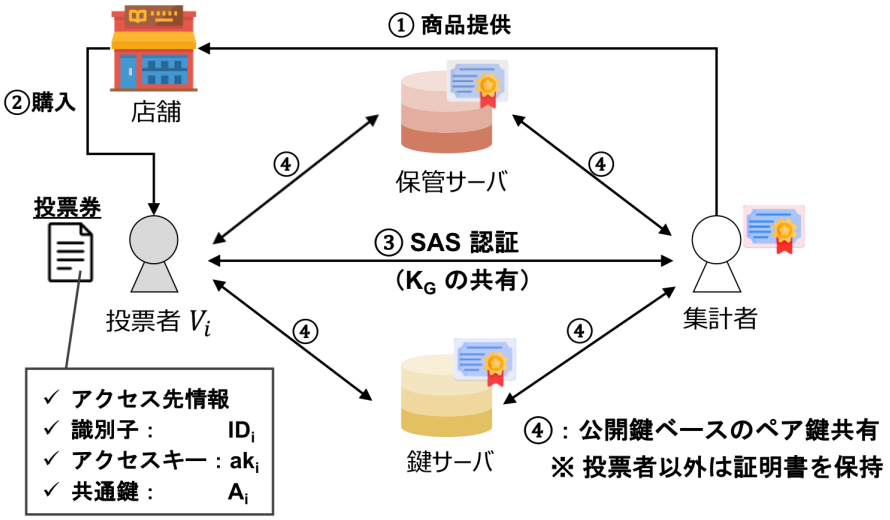

図 6.1　集計結果の検証

集計者が投票権を発行し，投票者は商品の購入で投票権を入手する．この投票権には，アクセス先情報，識別子，アクセスキー，集計者との共通鍵情報を含めておき，アクセス先情報と識別子，アクセスキーを用いて，集計者と SAS 認証を行う．つまり，投票権を入手した人のみ SAS における初回認証情報の登録が可能となる．SAS では，初回認証情報を安全に共有できれば，その後の認証は高速で安全に行うことができる．認証確立後，グループ鍵である K_G を共有する．投票者とその他のエンティティ間の鍵配送においては，公開鍵ベースのペア鍵共有法を用いることで安全に暗号鍵の共有を行うことができる．

今回のような投票権の入手にお金が発生する場合，投票の透明性が重要視される．提案方式では，集計者の不正を仮定した場合でも，鍵サーバによる集計結果の検証を実現しているため，単一の集計者による投票モデルよりも透明性が高いと言える．しかし，投票者の規模に応じた集計処理の負荷は今後検討していく必要がある．

6.5　考えられる脅威と対策

提案システムで特に問題となる脅威と対策について検討する．

6.5.1　マルウェアによる不正行為

投票者の端末や各サーバに，悪意のあるソフトウェアであるマルウェアが潜んでいると，投票行為の改ざんや集計結果の改ざんなど，多数の不正行為が考えられる．集計結果の改ざんに対しては，鍵サーバによる検証があるが，鍵サーバ自体がマルウェアに感染しているとどうしようもない．この対策として，保管サーバ，鍵サーバ，集計者端末にはセキュリティソフトをインストールしておき，投票を始める前に入念なチェックをしておくことが挙げられる．ただし，未知のマルウェアに対しては無力であるため，投票期間中は各サーバの挙動を随時監視しておく方が良い．投票者に関しては，投票者の利用端末を制御できる場合は，事前にセキュリティソフトのインストールが可能であるが，投票者の利用端末を制御できない場合は，セキュリティソフトの導入を勧める告知をするに留まる．そのため，マルウェア

による投票者の不正を完全に防ぐことはできない．

6.5.2　DDos 攻撃

多大なアクセスによりネットワークに負荷をかけることで，投票システムを停止させる DDos 攻撃 (Distributed Denial of Service attack) も想定される．提案システムでは，保管サーバと鍵サーバのどちらかが停止すると投票処理ができない．DDoS 攻撃への対策として，不正な投票者による頻繁なアクセスを防ぐため，同一 IP でのアクセス制限や WAF の利用などを検討する必要がある．

第 7 章

まとめ

本論文では，匿名情報収集方式の問題点として集計者の正当性を検証できないことを取り上げ，集計結果の正真性を保証するための検証プロトコルを提案した．投票者の代わりに鍵サーバが集計結果を検証するため，投票者の処理負荷は小さく抑えることができる．ただし，検証処理が可能となったことにより，鍵サーバの計算処理が増加しており，より効率の良い検証方法を検討する必要がある．また，具体的な適用箇所について，購入した商品に付属する投票権を用いた Web 上での投票を想定し，提案方式で考慮すべき点である，投票者と各サーバとの暗号鍵の共有方法について検討した．さらに，考えられる脅威についても，特に問題となりうる各端末のマルウェア感染や DDoS 攻撃について対策を検討した．

今後は，その他のエンティティによる不正を考慮し正当性を検証できること，効率の良い検証方法を検討すること，他の電子投票方式と安全性の比較評価を行い，その上で適用可能な投票規模を明確にすることが課題である．

謝辞

本研究の遂行と論文作成にあたって，ご指導，ご助言をいただきました高知工科大学情報学群 清水明宏副学長に心より感謝申し上げます．また，本研究の副査を担当していただきました高知工科大学情報学群 敷田幹文教授，福本昌弘教授に深く御礼申し上げます．

また，有益な議論を交わしていただいた高知工科大学 清水研究室の関係者各位に深く感謝致します．

参考文献

[1] 総務省. “電子機器利用による選挙システム研究会報告書.” http://www.soumu.go.jp/s-news/2002/pdf/020201_2.pdf, 閲覧日：2019 年 2 月 3 日.

[2] 税所哲郎, et al. “重み付き投票の電子化とその安全性に関する考察.” 情報処理学会論文誌, 44.8: 1913-1923, 2003.

[3] 谷川浩司, et al. “プライバシを保護した授業評価アンケートの実装.” 情報処理学会研究報告コンピュータセキュリティ (CSEC), 2006.81 (2006-CSEC-034): 375-381, 2006.

[4] 許容碩, et al. “票-取消と無証拠性が可能な不在者投票を含んだ電子投票の設計.” In: Proc. of the 2003 Symposium on Cryptography and Information Security. 電子情報通信学会情報セキュリティ研究専門委員会, p. 185-190. 2003.

[5] 高林茂樹. “二重鍵利用のインターネット投票.” 埼玉女子短期大学研究紀要= Bulletin of Saitama Women’s Junior College, 29: 11-22, 2014.

[6] 佐古和恵, 古川潤. “ミックスネットについて: 電子データをシャッフルする方法 (符号と暗号の代数的数理).” 2004.

[7] 石田夏樹, et al. “MIX-net と準同型性に基づいた電子投票方式.” 情報処理学会研究報告コンピュータセキュリティ (CSEC), 2004.75 (2004-CSEC-026): 335-342, 2004.

[8] 佐古和恵, 森健吾. “ミックスネットを用いた SCIS 論文賞電子投票実験について.” 電子情報通信学会 基礎・境界ソサイエティ Fundamentals Review, 2.1: 1_48-1_57, 2008.

[9] 赤羽喜治, 愛敬真生, “ブロックチェーン 仕組みと理論 サンプルで学ぶ FinTech のコア技術.” 株式会社リックテレコム, 2016.

[10] SCHNEIDER, Alexander. METER, Christian. HAGEMEISTER, Philipp. “Survey on Remote Electronic Voting.” arXiv preprint arXiv:1702.02798, 2017.

[11] 岡本龍明, 山本博資, “現代暗号.” 産業図書株式会社, 1997.

[12] 結城浩, “暗号技術入門　第三版.” SB クリエイティブ株式会社，2016.

[13] 安光穂高, “匿名情報収集方式の提案.” 高知工科大学学士学位論文，2017.

[14] T.Tsuji, T.Kamioka, A.Shimizu, “Simple And Secure password authentication protocol Ver.2(SAS-2).” IEICE Technical Reports, OIS2002-30, 2002.

[15] LIN, Chun-Li; SUN, Hung-Min; HWANG, Tzonelih. “Attacks and solutions on strong-password authentication.” IEICE transactions on communications, 84.9: 2622-2627, 2001.