6G-01

シャッフリングによる 大規模電子投票システムの実現

佐古 和恵† 古川 潤 † 森 健吾 † 尾花 賢 † 滝沢 政明†† 笹村 直樹†† 宮内 宏 †
†NEC 情報通信メディア研究本部 ††NEC 情報システムズ

概要

本稿では、シャッフリングにより無記名性を保証する安全な電子投票システムを実装したので報告する。大規模な投票者数にも耐えられるに様に設計した結果、千人の投票者に対してパソコンを用いて 80 秒で集計できた。万人規模の投票であっても、実用的な時間で集計できる見込みが得られた。

1 はじめに

投票を電子的に行なうことにより、集計にかかわる人件費を削減でき、高速で正確な結果が得られると、電子投票に対する期待は高まっている。さらに、ネットワーク上で投票者を確認できれば、指定された投票所に赴かなくても、任意の投票端末から投票することができ、投票者にとっての利便性が大幅に向上する。しかし、投票者確認と投票行為が連続すると、「誰がなにに投票したか」という投票の秘密が守れなくなってしまう。一方で、投票者確認は不正投票を防止するために必須の機能であり、投票行為と独立に行なうわけにはいかない。そこで、暗号プロトコルを用いて不正投票を防止しつつ、投票の秘密を守る方式が提案されている。さらには、正しい集計であることを誰でも確認できる方式も知られている。

このように、理論上では安全な投票システムを構築できることが知られているが、複雑な暗号プロトコルを、その機能が正しく実現されるように実装するのは容易ではない。誰もが集計の正しさを確認できる機能を備えた投票システムを実装したとの報告は未だ聞かれない。

筆者らは、シャッフリングに基づいた暗号プロトコルを用いて、数千人規模の投票にも耐えられるようなシステムを実装した。その結果、十分な強度をもったセキュリティパラメータを用いた場合でも、一千人の暗号投票文を約 80 秒で集計でき、正当性を検証する時間を含めても 2 時間ですべての処理が終了することがわかった。一万人の暗号投票文を約 15 分で集計できる目処がたった。なお、各有権者を認証する公開鍵基盤は別途存在するものとしている。

Realization of Large Scale Electronic Voting System using Shuffles
† Kazue Sako, Jun Furukawa, Kengo Mori, Satoshi Obana
†† Masaaki Takizawa, Naoki Sasamura
† Hiroshi Miyauchi
NEC C&C Media Research (†)
NEC Informatec Systems (††)

2 シャッフリングに基づく投票プロトコル

2.1 投票プロトコルの選定理由

不正投票、不正集計を防止しつつ投票の秘密をまもる投票プロトコルは、大きくわけて

分類 1 匿名通信路とブラインド署名技術を用いる方式 [O88, FOO92, Sak94]

分類 2 mix-net と呼ばれるシャッフリングを用いる方式 [Cha81, PIK93, SK95, Abe99]

分類 3 準同型性を持つ暗号化関数を用いる方式 [CY85, SK94, CGS97]

に分類できる。これらの特性を表 1 にまとめた。

[分類 1] の方式は計算量が少ないので、海外で実装が進んでいる [SENSUS, EVOX]。しかしながら、これらの方式は、各投票者が自分の票が集計されているかどうかは確認可能だが、第三者が投票全体の正当性を検証する手段はない。

[分類 2][分類 3] の方式は第三者が投票全体の正当性を検証できる手段を提供できるが、その分計算量が多くなる。また、[分類 3] の方式の多くはあらかじめ選択肢を固定する必要があり、汎用性、柔軟性に欠ける。

[分類 2] の方式は第三者による検証手段が提供でき、選択肢に自由度がある。計算量が多いものの、その大部分はセンタ側が負担するものであるから、投票側に必要となる計算量は小さい。そこで、本稿では [分類 2] の方式を採用した。

	第三者検証可能性	センタ計算量	投票者計算量	選択肢自由度	センタ数	投票者通信回数
分類 1	×	○	○	○	1	2
分類 2 (採用)	○	×	○	○	複数	1
分類 3	○	△	×	×	複数	1

表 1: 投票プロトコルの比較

2.2 投票プロトコルの概要

シャッフリングに基づく投票プロトコルの詳細は 4 章で述べるが、ここではその原理の概要を述べる。

m 個のシャッフルセンタ (SC) がそれぞれ個別の秘密鍵 D_i を所有する。図 1 には、$m = 3$ の例を示している。投票者はそれぞれのシャッフルセンタの公開鍵 E_i で多重に投票文を暗号化する。

$$E(vote) = E_1(E_2(\cdots(E_m(vote))))$$

投票者は投票センタの投票者確認を経た上で、暗号投票文を送信する。この時点で、投票センタは投票者名はわかるが、投票内容は暗号化されているので、投票の秘密は守られている。

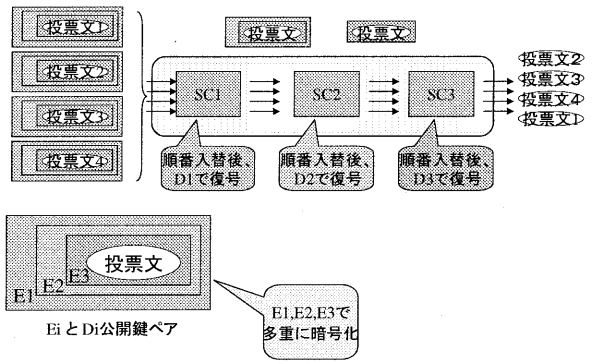

図 1: シャッフリングに基づくプロトコルの概要

投票センタが集めた暗号投票文のリストは、まず第一のシャッフルセンタ (SC 1) に届けられる。

第一のシャッフルセンタではまず、与えられた暗号データの順番をランダムに置換 (シャッフル) する。次に、シャッフルされた暗号データを自分の秘密鍵 D_1 を用いて復号する。このとき

$$D_1(E(vote)) = E_2(E_3(\cdots(E_m(vote))))$$

となる。すなわち、第一のシャッフルセンタの公開鍵で暗号化する前の状態が復元されるが、まだ第二シャッフルセンタ以降の公開鍵で暗号化されている状態である。その結果を次のシャッフルセンタに送る。次のシャッフルセンタも同様に入力をシャッフルしたのちに、それぞれのデータを自分の秘密鍵で復号し、次のシャッフルセンタに送る。このようにして、最後のシャッフルセンタが出力したものは、すべての暗号化がほどかれ可読な投票内容のリストになっている。

$$D_m(D_{m-1}(\cdots D_1(E(vote)))) = vote$$

投票センタは最後のシャッフルセンタの出力を集計し、投票結果を公表する。各シャッフルセンタがそれぞれにリストの順番を入れ替えているので、復号結果が入力のどの暗号文に対するものかは秘匿されている。

集計結果の正当性を示すためには、各シャッフルセンタが、正しく復号したことと正しくシャッフルしたことを、自分の秘密鍵やシャッフルした順番を漏らさずに証明すればよい。これはゼロ知識証明と呼ばれる手法を用いて実現できることが知られている。証明プロトコルの詳細は 4.4 節で述べる。

またエルガマル暗号を利用すれば、シャッフルセンタそれぞれの公開鍵で多重に暗号化処理を施さなくても、合成した公開鍵を用いて 1 度だけ暗号化すれば同値な効果が得られることも知られている [PIK93]。

なお、このように暗号化されたデータを複数のセンタがシャッフルしながら徐々に復号する方式は一般に mix-net と呼ばれている。

3 システム構成図及び投票の流れ

前述したシャッフリングに基づく暗号プロトコルを図 2 のようなシステム構成で実装した。

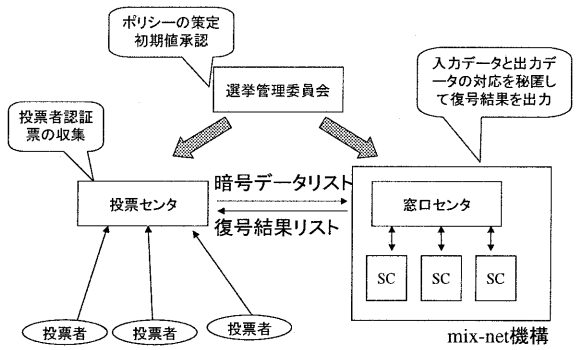

図 2: システム構成図

まず、選挙管理委員会が投票にかかわるポリシーを策定する。すなわち、投票議題や、有権者名簿、投票期日、投票センタ及び mix-net 機構の任命、プロトコルで用いられるセキュリティパラメータなどを決定する。

投票センタは選挙管理委員会が承認した有権者名簿を管理し、同様に承認された公開鍵情報を投票者向けに公開する。投票者はこの公開鍵を用いて、自分の投票文を暗号化し、投票センタに署名付きで送付する。投票センタは署名を検証し、投票者が有権者であり、まだ投票していないことを確認して暗号投票文を受理する。受理した暗号投票文に対して投票センタの署名つきレシートを発行し、投票者に返送する。

投票センタはきめられた期日で投票の受付を締切る。締切り後、全部の暗号投票データリストを mix-net 機構内の窓口センタに送付する。

窓口センタは投票センタの正当な依頼であることを確認し、複数のシャッフルセンタ (SC) を経由して暗号投票データリストの復号結果を投票センタに返送する。投票センタは復号結果に基づき票を集計し、集計結果を公開する。

また、mix-net 機構が正しく処理をしたことを示す証明も投票センタにて公開される。

4 集計プロトコルの詳細

本章では mix-net 機構に関する暗号プロトコルを中心に、公開鍵設定方法、投票文暗号化方法、集計方法、および集計プロセスの正当性証明方法について述べる。

4.1 公開鍵設定

エルガマル暗号のパラメータ

本プロトコルではエルガマル暗号を用いる [E85]。p, q はある整数 k により $p = kq + 1$ という関係にある 2 素数とする。g は法 p における位数 q の部分群を生成する生成元とする。

選挙管理委員会は、適切な p,q の長さを選定し、パラメータ (p,q,g) を具体的に選出する。あるいは、窓口センタに上記長さのパラメータの生成を依頼し、その正当性を確認した上で採用してもよい。

SC の公開鍵生成

選挙管理委員会は上記エルガマル暗号のパラメータ (p,q,g) をシャッフルセンタに通知する。$j(=1,\cdots,m)$ 番目の SC は、無作為に $x_j \bmod q$ を選び、秘密鍵とする。そして自分の公開鍵 y_j を

$$y_j = g^{x_j} \bmod p$$

により生成する。y_j に対して x_j を知っていることの証明 y'_j, r_j を

$$\begin{aligned} y'_j &= g^{\beta_j} \bmod p \\ c_j &= Hash(p,q,g,y_j,y'_j) \\ r_j &= c_j x_j + \beta_j \bmod q \end{aligned}$$

により生成する。そして、自分の公開鍵 y_j と証明文 y'_j, r_j を、署名つきで窓口センタに送付する。なお、ここで、$Hash$ は安全なハッシュ関数である。

公開鍵の合成

窓口センタはシャッフルセンタから送られた公開鍵 y_j の正当性を

$$\begin{aligned} c_j &= Hash(p,q,g,y_j,y'_j) \\ g^{r_j} {y_j}^{-c_j} &= y'_j \bmod p \\ {y_j}^q &= 1 \bmod p \\ y_j &\neq 1 \bmod p \end{aligned}$$

により検証する。検証された公開鍵を

$$Y = \prod_{j=1}^{m} y_j \bmod p$$

により合成して Y を得る。

選挙管理委員会は 各シャッフルセンタの公開鍵 y_j 及び合成された公開鍵 Y が正しく生成されていることを追確認し、公開鍵 Y を承認する。

4.2 投票文の暗号化

$i(=1,\cdots,n)$ 番目の投票者は、選挙管理委員会の承認した公開鍵 Y およびエルガマルパラメータ (p,q,g) を用いて投票内容 m_i を下記のように暗号化する。なお、ここで投票内容は位数が q になるように選ばれているものとする。

まず、投票者は任意の乱数 $r_i \bmod q$ を選ぶ。次に

$$(G_i, M_i) = (g^{r_i}, m_i Y^{r_i}) \bmod p$$

により得られた (G_i, M_i) を暗号文とする。この暗号文に署名をつけて、投票センタに送付する*。

* ここで、non-malleability を達成するために、投票者が m_i を知っていることの証明を施すことが望ましい。

4.3 集計開始

投票センタは正当な投票者 n 人からの暗号投票文をリスト $\{(G_i,M_i)\}_{(i=1,\cdots,n)}$ にして窓口センタに送付し、集計を依頼する。窓口センタは $\{G_i\},\{M_i\}$ の位数が q であることを確認してから、$(G_i,M_i) = (G_i^{(1)}, M_i^{(1)})(i=1,\cdots,n)$ とし、順次暗号データリスト $\{(G_i^{(j)}, M_i^{(j)})\}_{(i=1,\cdots,n)}$ を j 番目のシャッフルセンタに渡す。このシャッフルセンタから受け取る暗号データリストを $\{(G_i^{(j+1)}, M_i^{(j+1)})\}_{(i=1,\cdots,n)}$ として、次のシャッフルセンタに渡す。次にシャッフルセンタ内の処理について述べる。

シャッフルセンタのシャッフル

j 番目のシャッフルセンタはそれぞれの暗号文 $(G_i^{(j)}, M_i^{(j)})(i=1,\cdots,n)$ に対して、ランダムに置換写像 $\pi^{(j)}$ を決定し、それに基づいて $\{(G_i^{(j)}, M_i^{(j)})\}_{(i=1,2,\ldots,n)}$ の順番を入れ替え、新たに $\{(\bar{G}_i^{(j)}, \bar{M}_i^{(j)})\}_{(i=1,2,\ldots,n)}$ を得る。このとき、

$$\{(\bar{G}_i^{(j)}, \bar{M}_i^{(j)})\} = \{(G_{\pi^{(j)}(i)}^{(j)}, M_{\pi^{(j)}(i)}^{(j)})\}(i=1,2,\ldots,n)$$

が成り立っている。

シャッフルセンタの再暗号

次に、シャッフルセンタはシャッフルした暗号文を再暗号する。ここで再暗号とは、暗号化されたデータの内容を変えずに暗号文の「見かけ」を変えることである。単に位置を変えただけでは、暗号データのビットパターンから暗号文を追跡することが可能であるため、このような処理が必要になる。

そこで、j 番目のシャッフルセンタは自分以降のシャッフルセンタの公開鍵を合成して

$$Y_j = \prod_{\ell=j}^{m} y_\ell \bmod p$$

を求める。

そして、シャッフルされた $\{(\bar{G}_i^{(j)}, \bar{M}_i^{(j)})\}_{(i=1,2,\ldots,n)}$ に対して、乱数 $s_i^{(j)} \bmod q$ を発生させて

$$\begin{aligned} {G'}_i^{(j)} &= \bar{G}_i^{(j)} \cdot g^{s_i^{(j)}} \bmod p \\ {M'}_i^{(j)} &= \bar{M}_i^{(j)} \cdot Y_j^{s_i^{(j)}} \bmod p \end{aligned}$$

により $\{({G'}_i^{(j)}, {M'}_i^{(j)})\}_{(i=1,2,\ldots,n)}$ を求める。

シャッフルセンタの復号

上記のように、シャッフル、再暗号した $\{({G'}_i^{(j)}, {M'}_i^{(j)})\}_{(i=1,2,\ldots,n)}$ に対して秘密鍵情報 (x_j) を用いて

$${M''}_i^{(j)} = {M'}_i^{(j)} / ({G'}_i^{(j)})^{x_j} \bmod p$$

$$G''^{(j)}_i = G'^{(j)}_i$$

を計算する。そして、

$$(G''^{(j)}_i, M''^{(j)}_i)_{(i=1,2,\ldots,n)}$$

を窓口センタに渡す。

4.4 正当性証明および検証

ここで紹介する証明プロトコルは主に文献[SK95]に記載されているcut-and-choose証明法をベースにしたものである†。ただし、復号証明はSchnorr法[Schnorr]を利用して効率化を図った。なお、可読性のために、シャッフルセンタを特定する $^{(j)}$ の表記を省略する。

シャッフリング証明

入力暗号文リスト $\{(G_i, M_i)_{(i=1,\cdots,n)}\}$ に対して、置換写像 π を用いてシャッフルをし、公開鍵情報 (p, g, q, Y_j) と乱数列 $\{s_i\}_{(i=1,\cdots,n)}$ を用いて $\{(G'_i, M'_i)_{(i=1,\cdots,n)}\}$ を生成したことを、置換写像と乱数列を秘密にして証明するプロトコルを下記に示す。なお、ここで t は選挙委員会が決めたセキュリティパラメータである。

1. t 個のランダムな置換写像 π'_u と、$t \times n$ 個の乱数 $\{s'_{(u,v)}\}$ を発生させ、

$$A_{u,v} = G_{\pi'_u(v)} \cdot g^{s'_{(u,v)}} \bmod p$$
$$B_{u,v} = M_{\pi'_u(v)} \cdot Y_j^{s'_{(u,v)}} \bmod p$$

を $u = 1, \cdots t, v = 1, \cdots n$ に関して計算する。

2. $c = Hash(p||q||g||Y_j||\{(G_i, M_i)\}_{(i=1,\cdots,n)}||$
$\{(G'_i, M'_i)\}_{(i=1,\cdots,n)}||$
$\{(A_{(u,v)}, B_{(u,v)})_{(u=1,\cdots,t,v=1,\cdots,n)}\})$ を計算する。ここで $||$ は連結を示す。

3. c の u 番目 $(u = 1, \cdots, t)$ のビットを $c[v]$ とすると、

- $c[u] = 0$ のとき、$\alpha_u = \pi'_u$ 及び $\beta_{(u,v)} = s'_{(u,v)}$ とおく。
- $c[u] = 1$ のとき、$\alpha_u = \pi^{-1} \circ \pi'_u$ 及び $\beta_{(u,v)} = s'_{(u,v)} - s_{(\pi^{-1} \circ \pi'_v(u))}$ とおく。

証明文は $\{(G'_i, M'_i)_{(i=1,\cdots,n)}\}$, $\{(A_{(u,v)}, B_{(u,v)})\}$, $\{\alpha_u\}$,$\{\beta_{(u,v)}\}$ となり、シャッフルセンタはこれを公開する。

シャッフリング検証

検証は下記のように行なう。

1. 公開された情報及び証明文から
$c = Hash(p||q||g||Y_j||\{(G_i, M_i)\}_{(i=1,\cdots,n)}||$
$\{(G'_i, M'_i)\}_{(i=1,\cdots,n)}||$
$\{(A_{(u,v)}, B_{(u,v)})\}_{(u=1,\cdots,t,v=1,\cdots,n)})$
を計算する。

2. c の u 番目 $(u = 1, \cdots, t)$ のビットを $c[u]$ とすると、

- $c[u] = 0$ のとき、
$$A_{u,v} = G_{\alpha_u(v)} \cdot g^{\beta_{(u,v)}} \bmod p$$
$$B_{u,v} = M_{\alpha_u(v)} \cdot Y_j^{\beta_{(u,v)}} \bmod p$$
が成り立つことを確認する。
- $c[u] = 1$ のとき、
$$A_{u,v} = G'_{\alpha_u(v)} \cdot g^{\beta_{(u,v)}} \bmod p$$
$$B_{u,v} = M'_{\alpha_u(v)} \cdot Y_j^{\beta_{(u,v)}} \bmod p$$
が成り立つことを確認する。

すべてが確認できれば、この証明文は正しいとみなす。

復号証明及び検証

上記シャッフリングの証明の一部である $\{(G'_i, M'_i)_{(i=1,\cdots,n)}\}$ に対して自分の公開鍵 y に対する秘密鍵 x を用いて正しく計算した $\{(G''_i, M''_i)_{(i=1,\cdots,n)}\}$ を窓口センタに送ったことを秘密鍵を漏らさずに証明するプロトコルを下記に示す。

1. 乱数 $r \bmod q$ を生成する。
2. $c = Hash(p||q||g||y||\{G'_i\}_{(i=1,\cdots,n)}||$
$\{(M'_i/M''_i)\}_{(i=1,\cdots,n)}||g^r||\{G'^r_i\}_{(i=1,\cdots,n)})$
とおく。
3. $\alpha = r - cx \bmod q$ とおく。

以上のようにして求めた c, α が証明文となる。
検証は下記のように行なう。

1. すべての i について $G'_i = G''_i$ であることを確認する。
2. $c = Hash(p||q||g||y||\{G'_i\}_{(i=1,\cdots,n)}||$
$\{(M'_i/M''_i)\}_{(i=1,\cdots,n)}||g^\alpha \cdot y^c||$
$\{G'^\alpha_i \cdot (M'_i/M''_i)^c\}_{(i=1,\cdots,n)})$
が成り立つことを確認する。
3. すべての i について $M''^q_i = 1 \bmod p$ であることを確認する。

すべてが確認できれば、この証明文は正しいとみなす。

実装上の効率化

実際には、シャッフル証明と復号証明は一度に行なわれる。したがって、復号結果として与えられている $\{G''_i\}_{(i=1,\cdots,n)}$ に対して $G'_i = G''_i$ となるはずの $\{G'_i\}_{(i=1,\cdots,n)}$ を送る必要はない。したがって、実装ではシャッフル証明時に $\{M'_i\}_{(i=1,\cdots,n)}$ のみを送っている。

† 文献[MH96]に記載されている不備は解消されている。

5 実装上の考慮点

5.1 信頼モデル

本システム構成ではセンタと名のつくものが投票センタ、窓口センタ、シャッフルセンタと3種類ある。しかしながら、どのセンタも他のセンタあるいは一部の投票者と結託して不正をする可能性がある。

そこで、本システムでは選挙管理委員会というものを設け、ここで設定された投票ポリシーやセキュリティパラメータに基づいて各種センタの処理の正当性を評価するというモデルを採用した。選挙管理委員会は

- 投票センタ、窓口センタ、シャッフルセンタの任命
- 鍵長などのセキュリティパラメータの決定
- その他投票ポリシーの設定
- 窓口センタ及びシャッフルセンタが生成した公開鍵の認証

を行なう。

5.2 セキュリティを考慮した制御

論文などで紹介されている暗号プロトコルは、その本質的な性質のみが記述されており、他の技術でカバーできる既知の不正に対してはその記述が省略されがちである。しかし、実際は、そのような不正に対して採用する暗号プロトコルと相性の良い対策を選択して実装する必要がある。そこで、本システムの実装にあたっては、下記の点を配慮した。

まず、すべての通信にはデジタル署名を付与することにした。さらに、すべての通信にはログをとることにした。また、投票「セッション」に固有のセッション番号を付与した。同一のセッションに関して投票センタが窓口センタに復号を依頼すると、2度目は依頼を拒否されるような制御を加えた。同様に窓口センタが各シャッフルセンタに同じセッションに対して復号を2回以上依頼すると拒否される。なぜならば、ある暗号データリスト $(e_1,\ldots,e_m)$ と、これの j 番目の投票者のデータのみを変更した $(e_1,\ldots,e_{j-1},e',e_{j+1},\ldots,e_m)$ の両方をどちらもmix-net機構を通して復号したとする。そうすると、復号結果の差分から j 番目の投票者が何に投票しようとしたかわかってしまうからである。

なお、4.1節で記述したように、公開鍵の設定時にシャッフルセンタが「対応する秘密鍵を知っていることの証明」をする必要がある。それは他のシャッフルセンタの公開鍵 y_i を見て、合成される公開鍵 Y を自分の都合のいいように登録しないようにである。たとえば、悪意を持った ℓ 番目のシャッフルセンタが Y を $Y = g^X \bmod p$ となるように、自分の登録する公開鍵を

$$y_\ell = Y / \prod \text{残りのシャッフルセンタの公開鍵 } y_i$$

とすると、自分の知っている X を用いてすべての投票者の暗号投票文を復号できてしまう。ただし、このように生成した y_ℓ に対してその離散対数を計算することはできないので、秘密鍵を知っている証明を義務とすることでこのような不正を検出することができる。

また、4.1節及び4.4節において、公開鍵 y_j や復号結果 M_i'' の位数を確認している。これも確認を怠ることによって証明に不備が生じるので‡、注意を要する。

5.3 高速化

本実装では、文献[MM94]で報告した暗号ライブラリを用いている。このライブラリでは、剰余演算やべき乗剰余演算など単独の演算の高速化は達成されている。そこで、本システムのように多数のべき乗剰余演算を繰り返し行なう場合に有効となる高速化手段を模索した。

前章で記述したプロトコルは、1024ビットの p と160ビットの q を用い、セキュリティパラメータを160とすると、投票者数が1000の場合シャッフルセンタは正当性を証明するのに、321,000回ものべき乗剰余演算が必要になる。上記ライブラリでは法1024ビットのべき乗剰余演算は1回に約44msecの時間がかかる(CPU Pentium II 450MHz)ので、単純に計算すると1つのシャッフルセンタが正当性を証明するのに、4時間もかかることになる。

そこで、文献[MOV]からのsimultaneous multiple exponentiation および fixed-base windowing method for exponentiaion を採用し、高速化を図った。前者は $x^a \cdot y^b \bmod p$ の計算をほぼ1.2回のべき乗剰余演算程度の時間で実現するものである。後者は、同じ基底で繰り返し演算するときに有効となる事前計算テーブルを使うものである。この方式は、事前計算テーブルの大きさに関連するウインドウサイズと、実行するべき乗剰余演算の数、及び使用可能なメモリ領域の大きさが高速化に密接に関係がある。パラメータを変えて実測し最適なウインドウサイズを求めた。

以上のような最適化をはかった結果、上記の条件で1つのシャッフルセンタが約20分で正当性を証明できるようになった。

6 実装結果

5.3節で述べた高速化手法を使い、1024ビットの p と160ビットの q を用い、セキュリティパラメータを160、シャッフルセンタの数を3として実装した。その結果、1000人の暗号投票文を3つのシャッフルセンタがシャッフル、再暗号、復号、正当性証明をし、窓口センタがすべての証明を検証して集計結果を確定するのに窓口センタ、3つのシャッフルセンタそれぞれにPentium II 450MHzのパソコンを用いて2時間で実現できることがわかった。このうち、証明の生成や

‡ 復号証明が M_i' を c 乗することにより正当性を確認しているので、M_i' の位数が q でない場合にも証明が通ってしまう場合があるからである。

検証抜きで、集計結果だけを算出するのにかかった時間は約 80 秒であった。

この結果を 1 万人規模の投票に換算すると、同程度のパソコンを用いて 15 分程度で集計できることになる。より高速なパソコンを用いれば、現状でもこの時間は大幅に短縮できる。充分に大きなメモリを搭載することにより、事前計算テーブルを有効に活用できることもわかった。また、シャッフルセンタ内に並列 CPU を設ければ、さらなる高速化が見込める。

7 まとめ

シャッフリングを用いて安全な電子投票システムを設計し、大規模な投票にも耐え得るように実装した。その結果、3 つのシャッフルセンタを介して 1000 人分の暗号投票文を安全に約 80 秒で集計できることがわかった。1 万人規模の投票でも、15 分程度で集計できる見込みが得られた。

今後は文献 [F01] にあるようなより効率のよい証明プロトコルを実装したり、また、証明データの長さを削減するために楕円曲線暗号を用いるなどして、投票集計にかかるコストの一層の低減に務めたい。

参考文献

[Abe99] M. Abe: "Mix-networks on permutation networks," *Advances in Cryptology — ASIACRYPT '99*, pp. 258–273, Springer-Verlag, 1999.

[Cha81] D. Chaum: "Untraceable Electronic Mail, Return Addresses, and Digital Pseudonyms," *Communications of the ACM*, pp. 84–88, ACM, 1981.

[CY85] J. D. Cohen and M. Yung: "Distributing the power of a government to enhance the privacy of voters," *Annual Symposium on Principles of Distributed Computing*, pp. 52–62, 1985.

[CGS97] R. Cramer, R. Gennaro, and B. Schoenmakers: "A secure and optimally efficient multi-authority election scheme," *European Transactions on Telecommunications*, 8:481–489, 1997. Preliminary version in *Advances in Cryptology — EUROCRYPT '97.*

[E85] T. ElGamal: "A Public Key Cryptosystem and a Signature Scheme based on Discrete Logarithm," *IEEE Transactions on Information Theory,* Vol. IT-31, pp. 469–472, 1985.

[EVOX] http://theory.lcs.mit.edu/~cis/voting/voting.html

[FOO92] A. Fujioka, T. Okamoto, and K. Ohta: "A Practical Secret Voting Scheme for Large Scale Elections," *Advances in Cryptology – AUSCRYPT '92*, pp. 244–251, Springer-Verlag, 1992.

[F01] 古川 潤: 「効率の良い全体検証可能なシャッフル」, SCIS 2001, 15B-2

[MM94] 桝本, 宮内: 「可搬性の高い高速暗号演算ライブラリ」 第 49 回 情処全国大会論文集 1994.

[MOV] Menezes, van Oorschot, Vanstone: "Handbook of Applied Cryptography" CRC Press.

[MH96] M. Michels and P. Horster: "Some Remarks on a Receipt-Free and Universally Verifiable Mix-Type Voting Scheme," *Advances in Cryptology – ASIACRYPT '96*, pp. 125–132, Springer-Verlag, 1996.

[O88] 太田: 「単一の選挙管理者を用いた電子投票方式」電子情報通信学会春季全国大会 A-294",1-296, 1988.

[PIK93] C. Park, K. Itoh, and K. Kurosawa: "Efficient Anonymous Channel and All/Nothing Election Scheme," *Advances in Cryptology – EUROCRYPT '93,* pp. 248–259, Springer-Verlag, 1993.

[Sak94] K. Sako: "Electronic voting schemes allowing open objection to the tally," *Transactions of IEICE,* vol. E77-A No.1, Jan. 1994.

[SK94] K. Sako, J. Kilian: "Secure Voting using Partially Compatible Homomorphisms," *Advances in Cryptology – CRYPTO '94*, pp. 411–424, Springer-Verlag, 1994.

[SK95] K. Sako, J. Kilian: "Receipt-Free Mix-Type Voting Scheme," *Advances in Cryptology – EUROCRYPT'95*, pp. 393–403, Springer-Verlag, 1995.

[Schnorr] C. P. Schnorr: "Efficient signature generation by smart cards," *Journal of Cryptology*, 4, pp. 161–174, 1991.

[SENSUS] http://www.ccrc.wustl.edu/~lorracks/sensus/