社団法人　情報処理学会　研究報告
IPSJ SIG Technical Report

2004－ＣＳＥＣ－26　(48)
2004／7／21

集計時の負荷を軽減した重み付き電子投票プロトコル

中武　真治†　　中西　　透††　　舩曳　信生††　　杉山　裕二††

† 岡山大学大学院自然科学研究科
†† 岡山大学工学部

あらまし　重み付き投票とは，株主総会における議決権行使に代表されるように，1人が複数票を投票可能な投票である．従来，重み情報を秘匿できる重み付き電子投票プロトコルが提案されている．しかし，このプロトコルでは集計の際に各投票の再暗号化とランダムな置換を行わなければならないため，投票者数が多い場合，集計に時間を要する．そこで本稿では，集計時の負荷を軽減した重み付き電子投票プロトコルを提案する．
キーワード　重み付き投票，ElGamal暗号，知識の署名

An Electronic Weighted Voting Protocol with Tallying Cost Reduced

Shinji NAKATAKE†, Toru NAKANISHI††, Nobuo FUNABIKI††, and Yuji SUGIYAMA††

† Graduate School of Natural Science and Technology, Okayama University
†† Faculty of Engineering, Okayama University

Abstract　A weighted voting is a system such that a voter submits a vote with a weight, e.g., voting in stockholders' meetings. A weighted voting protocol that can keep weights secret has been proposed. However, this protocol has a problem that a tallying cost is large when there are many voters, since a management server must re-encrypt and permute all votes. In this paper, we propose a weighted voting protocol with tallying cost reduced.
Key words　weighted voting, ElGamal encryption, signature of knowledge

1. まえがき

従来，暗号技術を利用した電子投票プロトコルが数多く提案されているが，その多くでは，1人が1票のみ投票可能となっている．しかし，1人が複数票を投票可能な投票（重み付き投票）も存在する．例えば，株主総会における議決権行使が挙げられる．また，国連安保理の投票など，ある一定の条件を課せられた投票は重み付き投票を用いて実現できることが知られている[1]．

本稿では，投票対象は1つで，投票者は賛成（投票内容1）または反対（投票内容-1）のいずれかの値を重み分だけ投票すると仮定する．そして，重みを含めて集計した結果，賛成総数が反対総数以上の値になった場合に可決と扱うことにする．重みを分割して投票する方式も提案されている[2]が，ここでは[1]と同様に分割不可としている．一方，投票では，プライバシー保護のために投票内容は秘匿されるべきである．さらに，株主総会の例では各投票者の重みも秘匿されるべきである．なぜなら，一部の大株主を除いて株主の所有株式数は個人情報であり，公開されるべきではないためである．

以降で扱う投票プロトコルでは，投票者の重みを保証する信頼のおける管理サーバの存在を仮定する．株主総会の例では，信託銀行がこれに相当する．また，複数の集計サーバによって集計を行うこととする．これは，不正な集計を防止するためである．このとき，ある閾値以上の集計サーバが不正を行うことはないと仮定する．

文献[1]では，上記の状況において，重み情報を秘匿できる重み付き電子投票プロトコルが提案されている．このプロトコルでは，平文m_1, m_2に対して，$E(m_1) \times E(m_2) = E(m_1 + m_2)$を満たす準同型暗号$E$を利用する．各投票者は，準同型暗号を用いて1（賛成）または-1（反対）の暗号文を，投票として公開掲示板に掲示する．管理サーバは，各投票E_iを再暗号化し，ランダムな置換πで入れ替えた$E'_{\pi(i)}$を，対応する重みと一緒に$(E'_{\pi(i)}, w_{\pi(i)})$として掲示する．さらに，置換のみを行っていることを証明する．これにより，E_iとw_iを関連づけできなくなるので，重み情報を秘匿できる．一方，各投票者の重みの正しさも保証される．次に，各投票者の${E'_{\pi(i)}}^{w_{\pi(i)}}$をすべて乗ずる．準同型暗号$E$による暗号文を乗算すると，平文の投票内容自体は加算されるので，結果は（賛成総数）$-$（反対総数）の暗号文になる．これを集計サーバが復号し，可決か否かを示すことができる．

しかし[1]のプロトコルでは，集計の際に各投票の再暗号化とランダムな置換を行い，その正当性を証明しなければならな

い．これらの処理に要する時間は投票者の数に比例するので，投票者数が多い場合，集計に時間を要する．

そこで本稿では，集計に要する時間を軽減した重み付き電子投票プロトコルを提案する．提案プロトコルでは，1または-1の暗号文ではなく，w_iまたは$-w_i$の暗号文を投票する．まず，投票者V_iは事前に管理サーバから重みw_iを保証した署名を入手しておく．次に，準同型暗号を利用したw_i（賛成）または$-w_i$（反対）の暗号文を，投票として公開掲示板に掲示する．このとき，暗号化されている値がw_iまたは$-w_i$であることに加えて，そのw_iが管理サーバの保証した重みであることを零知識証明する．これにより，管理サーバによる置換の処理なしで，不正な重みによる投票を防止できる．集計時には，各投票者の投票内容の暗号文をすべて乗じて，（賛成総数）$-$（反対総数）の暗号文を生成し，これを集計サーバが復号して，可決か否かを示す．従来プロトコルの集計において，集計時間の大部分は，各投票の再暗号化とランダムな置換によるものである．提案プロトコルでは，集計の際にこれらの処理を行う必要がないため，従来プロトコルと比較して短い時間で集計を行うことができる．

2. 重み付き投票のモデルと要件

2.1 モデル

本稿で扱う重み付き投票プロトコルには以下が参加する[1]．

投票者V_i： 投票を行うN人の投票者．

管理サーバA： 投票者の重みを保証する信頼のおけるサーバ．

集計サーバT_i： 投票の集計を行うサーバ．複数のサーバにより構成され，ある指定された閾値以上のサーバは結託しないものとする．

2.2 投票の要件

重み付き投票の電子化を考える場合に満たすべき要件を以下に示す[1]．従来法[1]と同様に不統一行使防止を仮定する．

完全性： 不正がなければ正しく集計される．

無記名性： 投票者と投票内容の関連付けに関して，投票結果のみから得られる以上の情報を得ることができない．

2重投票不可能性： 投票者は2度投票することはできない．

未登録者の投票排除： 登録された投票者以外は投票できない．

公平性： 投票に影響する途中経過が露呈しない．

検証性： 投票結果が正しいことを誰でもが検証できる．

不統一行使防止： 投票者の投票内容はすべて同一である．すなわち，投票者の投票できる内容が，重みw_iの賛成，または重みw_iの反対のいずれか一方に限られる．

重み情報の秘匿： 投票者の重みに関して，投票結果のみから得られる以上の情報を得ることができない．

3. 利用する暗号技術

ここでは，提案プロトコルで用いるCamenischらによる署名，ElGamal暗号，知識の署名について述べる．以降，$[a,b]$は$a \leqq x \leqq b$となるようなすべての整数xからなる範囲，(a,b)は$a < x < b$となるようなすべての整数xからなる範囲を表す．また，$\in_R$はある集合からランダムに要素を選ぶことを表す．

3.1 Camenischらによる署名

提案プロトコルでは，Camenischらによる署名[3]を利用しており，その安全性は強RSA仮定の下で示されている．この仮定を説明した上で署名方式の概要を述べる．

ℓ_nをセキュリティパラメータとする．$n=pq$が安全なRSAの法である（即ち，$p=2p'+1, q=2q'+1$においてp,q,p',q'が素数である）場合，位数$p'q'$の要素によって生成されたZ_n^*の巡回部分群QR_n（平方剰余数からなるZ_n^*の部分群）は以下の強RSA仮定を満たす．

定義1（強RSA仮定）

RSAの法nと$u \in QR_n$が与えられたとき，$v^e = u \pmod n$を満たす$v \in QR_n$と$e \in Z_{>1}$を無視できない確率で出力することができる確率的多項式時間アルゴリズムは存在しない．

この署名方式は以下のアルゴリズムにより構成される．

鍵生成アルゴリズム：

$\ell_e \geqq \ell_m + 2, \ell_s = \ell_n + \ell_m + \ell$となるようなセキュリティーパラメータ$\ell_n, \ell_m, \ell_e, \ell_s, \ell$を設定する（$\ell \approx 128$）．そして長さ$\ell_n$のRSAの法$n$を計算し，$a,b,c \in_R QR_n$を選び，これらを公開する．また，$n$の素因数を秘密鍵として保持する．

署名アルゴリズム：

メッセージ$m \in [0, 2^{\ell_m})$に対して，素数$e \in_R (2^{\ell_e-1}, 2^{\ell_e})$と$s \in_R [0, 2^{\ell_s})$を選ぶ．そして$v$を以下のようにして計算する．

$$v = (a^m b^s c)^{1/e}$$

このとき，メッセージmに対する署名は(s,e,v)となる．

検証アルゴリズム：

(s,e,v)がメッセージmに対する署名であるかどうかを検証するために，以下が成り立つかを調べる．

$$v^e = a^m b^s c \ \wedge \ e \in (2^{\ell_e-1}, 2^{\ell_e})$$

3.2 ElGamal暗号

提案プロトコルでは，QR_n上のElGamal暗号を用いる[4]．ここでは，以下の記法を用いる．

g	:	QR_nの生成元．
$x \in Z_n$	:	秘密鍵．
y	:	$y = g^x$を満たす公開鍵．

このとき平文mの暗号文$E(m)$は，乱数$r \in Z_n$に対して以下のようになる．

$$E(m) = (g^r, y^r m)$$

また，復号は以下の通りとなる．

$$\frac{y^r m}{(g^r)^x} = m$$

ElGamal暗号文$E(m) = (g^r, y^r m)$に対して，演算$\otimes$, $^{-1}$と冪乗（α乗）を以下のように定義する．

$$E(m_1) \otimes E(m_2) = (g^{r_1} g^{r_2}, (y^{r_1} m_1)(y^{r_2} m_2))$$

$$E(m)^{-1} = ((g^r)^{-1}, (y^r m)^{-1})$$
$$E(m)^{\alpha} = ((g^r)^{\alpha}, (y^r m)^{\alpha})$$

このとき，$E(m_1) \otimes E(m_2) = E(m_1 m_2)$，$E(m)^{-1} = E(m^{-1})$，$E(m)^{\alpha} = E(m^{\alpha})$ となり，乗法において準同型性を満たす．

提案プロトコルでは，複数のサーバによる閾値 ElGamal 復号 [5] を行う．このため，複数のサーバは復号鍵を分散共有し，閾値以上のサーバが協力することで暗号文を復号する．

3.3 知識の署名

知識の署名 (SPK) [6] とは，知識の零知識証明を変換することにより得られる署名である．知識の零知識証明とは，証明者が秘密情報自体を明かすことなく，その秘密情報を知っていることのみを，検証者との相互通信によって証明する方式である．そして，この証明を耐衝突ハッシュ関数を用いて署名化したものが知識の署名である．証明者が述部 $Predicates$ を満たす秘密情報 $\alpha, \beta, \cdots$ を知っていることを証明するメッセージ m に対する署名を以下のように記述する．

$$SPK\{(\alpha, \beta, \cdots) : Predicates\}(m).$$

この表記では，ギリシャ文字は証明される知識の要素を表し，他の文字は署名者と検証者に公開されたパラメータを表す．このとき，RSA の法 n に対して以下の QR_n 上の SPK を利用する [3] [6] [7]．各 SPK の詳細は付録に示す．ここで，$\ell(\approx 160)$ をセキュリティパラメータとする．また，各証明される秘密はある正整数 ℓ_x に対して $[-2^{\ell_x}, 2^{\ell_x}]$ から選ばれるとする．

複数のリプリゼンテーションの知識を証明する SPK：

底 $g_1, \cdots, g_k$ に対する公開鍵 $y_1, \cdots, y_n$ のリプリゼンテーションの SPK を

$$SPK\{(\alpha_{ij})_{i=1,\cdots,n;j\in\mathcal{J}_i} : \bigwedge_{i=1}^{n} y_i = \prod_{j\in\mathcal{J}_i} g_j^{\alpha_{ij}}\}(m)$$

と表記する．ここで，$\mathcal{J}_i \subseteq \{1, \cdots, k\}$ は $y_i = \prod_{j\in\mathcal{J}_i} g_j^{\alpha_{ij}}$ が成り立つような集合である．

リプリゼンテーションの成分の等しさを証明する SPK：

底 $g_1, \cdots, g_k$ に対する公開鍵 $y_1, \cdots, y_n$ のリプリゼンテーションの成分の一部が等しいことを証明する SPK を

$$SPK\{(\alpha_1, \cdots, \alpha_u) : (y_1 = \prod_{j\in\mathcal{J}_1} g_j^{\alpha_{e_{1j}}}) \wedge \cdots \wedge (y_n = \prod_{j\in\mathcal{J}_n} g_j^{\alpha_{e_{nj}}})\}(m)$$

と表記する．ここで，インデックス $e_{ij} \in \{1, \cdots, u\}$ は秘密 $\alpha_1, \cdots, \alpha_u$ を参照する．

いくつかの公開鍵のうちの一つのリプリゼンテーションの知識を証明する SPK：

底 $g_1, \cdots, g_k$ に対する公開鍵 $\{y_1, \cdots, y_n\}$ のうちの（少なくとも）一つの y_i のリプリゼンテーションの知識を証明する SPK を

$$SPK\{(\alpha_{ij})_{i=1,\cdots,n;j\in\mathcal{J}_i} : \bigvee_{i=1}^{n} y_i = \prod_{j\in\mathcal{J}_i} g_j^{\alpha_{ij}}\}(m)$$

と表記する．

リプリゼンテーションの一部がある範囲に存在することを証明する SPK：

底 $g_1, \cdots, g_k$ に対する公開鍵 y のリプリゼンテーションの一部が，ある整数 a と正整数 b に対して，範囲 $[a - 2^{\tilde{\ell}}b, a + 2^{\tilde{\ell}}b]$ に存在することを証明する SPK を

$$SPK\{(\alpha_1, \cdots, \alpha_k) : y = \prod_{i=1}^{k} g_i^{\alpha_i} \wedge \alpha_i \in [a - 2^{\tilde{\ell}}b, a + 2^{\tilde{\ell}}b]\}(m)$$

と表記する．ただし，実際には α_i は $[a, a+b]$ の範囲になければならない．

4. 提案プロトコル

4.1 プロトコル

鍵の生成：

管理サーバ A は長さが $\ell_n/2$ の素数 $p = 2p' + 1$，$q = 2q' + 1$ (p', q' は素数) を選んで，長さが ℓ_n の RSA の法 $n = pq$ を生成する．そして，$a, b, c, g, h \in_R QR_n$ を選び，公開鍵 $PK = (n, a, b, c, g, h)$ と秘密鍵 $SK = p$ を出力する．

各集計サーバ T_i は乱数 x_i を秘密に生成する．次に，閾値型 ElGamal 暗号の公開鍵 $y = g^x$ を生成する [5]．このとき，公開鍵は y，秘密鍵は x となる．

重みを保証する署名の発行：

管理サーバ A は，以下のようにして，投票者の重みを保証する署名を発行する．

(Step 1) A は，$\ell_e \geq \ell_w + 2$，$\ell_s = \ell_n + \ell_w + \ell$ を満たすようなセキュリティパラメータ $\ell_w, \ell_s, \ell_e, \ell$ を設定する．また，これらのパラメータを基に以下の値の範囲を設定する．

$$\mathcal{W} = [0, 2^{\ell_w}),\ \mathcal{E} = (2^{\ell_e - 1}, 2^{\ell_e})$$

ある値がこれらの範囲に存在することを証明する際には，実際にはより狭い範囲から値を選ぶ必要がある（3.3 節参照）．範囲の証明によって $\mathcal{W}, \mathcal{E}$ の範囲に拡張される範囲をそれぞれ以下のように $\tilde{\mathcal{W}}, \tilde{\mathcal{E}}$ と記す．

$$\tilde{\mathcal{W}} = [2^{\ell_w - 1}, 2^{\ell_w - 1} + 2^{\ell_w - 2 - \tilde{\ell}}]$$

$$\tilde{\mathcal{E}} = [2^{\ell_e - 1} + 2^{\ell_e - 2}, 2^{\ell_e - 1} + 2^{\ell_e - 2} + 2^{\ell_e - 3 - \tilde{\ell}}]$$

また，投票者 V_i の重み $w_i \in [0, 2^{\ell_w - 2 - \tilde{\ell}}]$ に対して，$\tilde{w}_i = w_i + 2^{\ell_w - 1}$ とする．このとき，$\tilde{w}_i \in \tilde{\mathcal{W}}$ である．A は重み情報 $\tilde{w}_i \in \tilde{\mathcal{W}}$ に対して，素数 $e \in_R \tilde{\mathcal{E}}$ と $s \in_R [0, 2^{\ell_s})$ を選び，$v^e \equiv a^{\tilde{w}_i} b^s c \bmod n$ のような値 v を計算する．

(Step 2) A は，重み w_i を保証する署名として，s, e, v を投票者 V_i に送信する．

準備：

投票前の準備として，投票者 V_i は，マスク値（w_i または $-w_i$）の暗号文と，そのマスク値の w_i が管理サーバによって保証された重みであることを証明する SPK を生成し，公開掲示板に掲示する．集計サーバは SPK の正当性を検証する．

1. 投票者の処理

(Step 1) 投票者 V_i は乱数 $u_i \in_R \{w_i, -w_i\}$ をマスク値として選び，さらに乱数 $r \in_R [0, 2^{\ell_n})$ を選んで閾値型 ElGamal 暗号による h^{u_i} の暗号文を生成する．暗号文は $(C_{1i}, C_{2i}) = (g^r,\ y^r h^{u_i})$ となる．

(Step 2) (C_{1i}, C_{2i}) が h^{w_i} または h^{-w_i} の暗号文であり，w_i は A が保証した重みであることを零知識証明する以下の SPK を生成する．ここで，m は投票を区別する番号と投票者の ID 番号を組み合わせた値とする．

$$
\begin{aligned}
&SPK\{(\gamma, \omega, \epsilon, \sigma):\\
&(C_{1i} = g^{\gamma} \wedge C_{2i} h^{2^{\ell_w - 1}} = y^{\gamma} h^{\omega}\\
&\wedge c = v^{\epsilon} (1/a)^{\omega} (1/b)^{\sigma}\\
&\wedge \omega \in \mathcal{W} \wedge \epsilon \in \mathcal{E})\\
&\vee (C_{1i} = g^{\gamma} \wedge C_{2i} (1/h)^{2^{\ell_w - 1}} = y^{\gamma} (1/h)^{\omega}\\
&\wedge c = v^{\epsilon} (1/a)^{\omega} (1/b)^{\sigma}\\
&\wedge \omega \in \mathcal{W} \wedge \epsilon \in \mathcal{E})\}(m)
\end{aligned}
$$

$u_i = w_i$ の場合，以下のようにして，この SPK を生成する．ここで，ℓ_c をセキュリティパラメータとし，$\mathcal{H}$ を耐衝突ハッシュ関数（$\mathcal{H}: \{0,1\}^* \to \{0,1\}^{\ell_c}$）とする．

(Step 2-1) まず，$c_2 \in_R \{0,1\}^{\ell_c}$ を選ぶ．次に，$r_{11}, s_{21} \in_R [0, 2^{\ell_n + \tilde{\ell}})$，$r_{12} \in_R [0, 2^{\ell_w - 2})$，$s_{22} \in_R [c_2 2^{\ell_w - 2 - \tilde{\ell}}, 2^{\ell_w - 2})$，$r_{13} \in_R [0, 2^{\ell_c - 3})$，$s_{23} \in_R [c_2 2^{\ell_c - 3 - \tilde{\ell}}, 2^{\ell_c - 3})$，$r_{14}, s_{24} \in_R [0, 2^{\ell_s + \tilde{\ell}})$ を選ぶ．

(Step 2-2) 以下の値を計算する．

$$
\begin{aligned}
t_{11} &:= g^{r_{11}}\\
t_{12} &:= y^{r_{11}} h^{r_{12}}\\
t_{13} &:= v^{r_{13}} (1/a)^{r_{12}} (1/b)^{r_{14}}\\
t_{21} &:= g^{s_{21}} C_{1i}^{-c_2}\\
t_{22} &:= y^{s_{21}} (1/h)^{s_{22} + c_2 2^{\ell_w - 1}} (C_{2i} (1/h)^{2^{\ell_w - 1}})^{-c_2}\\
t_{23} &:= v^{s_{23} + c_2 (2^{\ell_c - 1} + 2^{\ell_e - 2})}\\
&\quad (1/a)^{s_{22} + c_2 2^{\ell_w - 1}} (1/b)^{s_{24}} c^{-c_2}\\
c_1 &:= \mathcal{H}(g||y||h||(1/h)||v||(1/a)||(1/b)||\\
&\quad C_{1i}||C_{2i} h^{2^{\ell_w - 1}}||c||C_{1i}||C_{2i}(1/h)^{2^{\ell_w - 1}}||c\\
&\quad ||t_{11}||t_{12}||t_{13}||t_{21}||t_{22}||t_{23}||m) \oplus c_2
\end{aligned}
$$

$$
\begin{aligned}
s_{11} &:= r_{11} + c_1 r\\
s_{12} &:= r_{12} + c_1 (\tilde{w}_i - 2^{\ell_w - 1})\\
s_{13} &:= r_{13} + c_1 (e - (2^{\ell_e - 1} + 2^{\ell_e - 2}))\\
s_{14} &:= r_{14} + c_1 s
\end{aligned}
$$

これにより，SPK として $(c_1, c_2, s_{11}, s_{12}, s_{13}, s_{14}, s_{21}, s_{22}, s_{23}, s_{24})$ を得る．

$u_i = -w_i$ の場合も，同様にして SPK を生成する．

(Step 3) h^{u_i} の暗号文 (C_{1i}, C_{2i})，SPK，v を公開掲示板に掲示する．

2. サーバの処理

集計サーバは，(C_{1i}, C_{2i}) が h^{w_i} または h^{-w_i} の暗号文であり，w_i が管理サーバの保証した重みであるかどうかを検証する．具体的には，以下の式が成立するかどうかを検証し，不成立の場合はその暗号文を排除する．

$$
\begin{aligned}
c_1 \oplus c_2 = \mathcal{H}&(g||y||h||(1/h)||v||(1/a)||(1/b)||\\
&C_{1i}||C_{2i} h^{2^{\ell_w - 1}}||c||C_{1i}||C_{2i}(1/h)^{2^{\ell_w - 1}}||c||\\
&g^{s_{11}} C_{1i}^{-c_1}||y^{s_{11}} h^{s_{12} + c_1 2^{\ell_w - 1}} (C_{2i} h^{2^{\ell_w - 1}})^{-c_1}\\
&||v^{s_{13} + c_1 (2^{\ell_e - 1} + 2^{\ell_e - 2})} (1/a)^{s_{12} + c_1 2^{\ell_w - 1}}\\
&(1/b)^{s_{14}} c^{-c_1}||g^{s_{21}} C_{1i}^{-c_2}||y^{s_{21}} (1/h)^{s_{22} + c_2 2^{\ell_w - 1}}\\
&(C_{2i}(1/h)^{2^{\ell_w - 1}})^{-c_2}||v^{s_{23} + c_2 (2^{\ell_e - 1} + 2^{\ell_e - 2})}\\
&(1/a)^{s_{22} + c_2 2^{\ell_w - 1}} (1/b)^{s_{24}} c^{-c_2}||m)\\
\wedge\ & s_{12} \in [c_1 2^{\ell_w - 2 - \tilde{\ell}}, 2^{\ell_w - 2})\\
\wedge\ & s_{13} \in [c_1 2^{\ell_e - 3 - \tilde{\ell}}, 2^{\ell_e - 3})\\
\wedge\ & s_{22} \in [c_2 2^{\ell_w - 2 - \tilde{\ell}}, 2^{\ell_w - 2})\\
\wedge\ & s_{23} \in [c_2 2^{\ell_e - 3 - \tilde{\ell}}, 2^{\ell_e - 3})
\end{aligned}
$$

投票：

投票者 V_i は，$e_i u_i = v_i \in \{w_i, -w_i\}$ となる $e_i \in \{1, -1\}$ を公開掲示板に掲示する．ここで v_i とは，投票者 V_i の投票内容である．

集計：

集計サーバは，以下のようにして集計を行う．

(Step 1) $(\tilde{C}_1, \tilde{C}_2) = (\prod C_{1i}^{e_i}, \prod C_{2i}^{e_i})$ を公開掲示板上で計算する．

(Step 2) 各集計サーバ T_i は部分情報 $a_i \equiv \tilde{C}_1^{\ x_i}$ を公開する．そして，a_i が正しいことを証明する．閾値以上の部分情報が集まれば，誰でも $\tilde{C}_1^{\ x}$ を計算できるので，復号して h^M を得ることができる [5]．ここで，M は（賛成総数）−（反対総数）である．

(Step 3) M が小さいことから，h^M の離散対数問題を解くことによって M を求める．

4.2 安 全 性

完全性： 各投票者が掲示した暗号文 $(C_{1i}, C_{2i}) = (g^r,\ y^r h^{u_i})$ を e_i 乗することによって，$(C_{1i}^{e_i}, C_{2i}^{e_i}) = (g^{re_i},\ y^{re_i} h^{u_i e_i}) = (g^{re_i},\ y^{re_i} h^{v_i})$ のようにして，各投票者 V_i の投票内容 v_i の暗号文 $(C_{1i}^{e_i}, C_{2i}^{e_i})$ が得られる．この暗号文は準同型性を持つので，すべての投票者の暗号文を掛け合わせることによって，$(\tilde{C}_1, \tilde{C}_2) = (\prod C_{1i}^{e_i}, \prod C_{2i}^{e_i}) = (g^{\sum re_i},\ y^{\sum re_i} h^{\sum v_i})$ のようにして，すべての投票者の投票内容の和（賛成総数 − 反対総数）$M = \sum v_i$ に対応した暗号文 $(\tilde{C}_1, \tilde{C}_2)$ を生成できる．$(\tilde{C}_1, \tilde{C}_2)$ を復号すると M を得ることができるので，不正がなければ正しく集計される．

無記名性： 投票内容は ElGamal 暗号によって暗号化されており，暗号化されたまま集計されるので，暗号文から投票内容が漏れることはない．また，準備の（Step 2）で生成する SPK は零知識証明であり，投票で送信する e_i は乱数なので，これらの情報からも投票内容が漏れることはない．よって，投票者と投票内容の関連付けに関して，投票結果のみから得られる情報以上の情報を得ることはできない．

2 重投票不可能性： 投票者の ID 番号が公開されるので，投票者は 2 度投票することはできない．

未登録者の投票排除： 投票者の ID 番号が公開されるので，未登録者は投票できない．

公平性： 無記名性と同様の理由により，投票者の情報から投票内容が漏れることはない．さらに，集計結果は投票が締め切られた後に公表される．以上のことから，投票に影響する途中経過は露呈しない．

検証性： 各投票者は，投票内容が w_i または $-w_i$ のどちらかであり，w_i が管理サーバの署名した重みであることを証明しなければならない．また，その署名は偽造することができない．したがって，各投票者は w_i または $-w_i$ のいずれか一方の値しか投票できない．なお，SPK のメッセージ m には，投票を区別する番号と投票者の ID が含まれているので，リプレイを行うこともできない．また，投票内容の暗号文は公開掲示板上で公開されているので，各投票者の投票内容の暗号文が正しく集計されていることを誰でも検証できる．また，各集計サーバは復号に必要な部分情報 $\tilde{C}_1^{x_i}$ を公開し，その正当性を証明しているため，閾値以上の部分情報が集まれば，誰でも閾値型 ElGamal 暗号文を正しく復号して h^M を得ることができる．さらに，M が小さいことから $h^M \bmod p$ の離散対数問題を解くことによって M を求めることができる．以上のことから，誰でも投票結果が正しいことを検証できる．

不統一行使防止： 各投票者は，投票内容が w_i または $-w_i$ のどちらかであり，w_i が管理サーバの署名した重みであることを証明しなければならない．したがって，各投票者は w_i または $-w_i$ のいずれか一方の値しか投票できない．

重み情報の秘匿： 無記名性と同様の理由により，暗号文，SPK，e_i から重み情報が漏れることはない．さらに，準備の（Step 3）で掲示する v から重みを求めることもできない．なぜなら，重み w_i はリプリゼンテーションの一部であり，リプリゼンテーションを求めることは困難であるためである．以上のことから，投票者の重みに関して，投票結果のみから得られる情報以上の情報を得ることはできない．

4.3 効　　率

従来法の集計では，全投票者の暗号文に対して再暗号化とランダムな置換を行い，その正当性を証明しなければならない．これらの処理には，投票者数 N に対して $\mathcal{O}(N)$ の冪乗剰余演算が必要となる．提案プロトコルの集計では，これらの処理を行う必要がないため，従来法と比較して短い時間で集計を行うことができる．

次に，投票前の準備段階における処理について見ていく．従来法では，暗号文が 1 または -1 を暗号化したものであることを検証する際に，底が 2 つの冪乗剰余演算を 4 回行わなければならない．ここで，複数の底が存在する場合の冪乗剰余演算の計算量について次のように考える．まず，底が 1 つの冪乗剰余演算の計算量を 1 とする．このとき，事前計算することによって，底が 2 つの冪乗剰余演算を約 1.2，底が 3 つの冪乗剰余演算を約 1.25，底が 4 つの冪乗剰余演算を約 1.31 の計算量で処理することができる．したがって，従来法では約 $4.8N$ 回の冪乗剰余演算が必要となる．一方，提案プロトコルでは，各投票者 V_i の暗号文が w_i または $-w_i$ を暗号化したものであり，w_i が管理サーバの保証した重みであることを検証する際に，底が 2 つの冪乗剰余演算，底が 3 つの冪乗剰余演算，底が 4 つの冪乗剰余演算をそれぞれ 2 回行わなければならない．したがって，約 $7.52N$ 回の冪乗剰余演算が必要となる．このように，提案プロトコルでは投票前の準備段階における計算量が増加しているが，十分に許容できる範囲である．

5. むすび

本稿では，集計時の負荷を軽減した重み付き電子投票プロトコルを提案した．今後の課題としては，一度に複数議案について投票できる方式への拡張や，選択肢が多数存在するような投票への対応が考えられる．

文　　献

[1] 税所哲朗, 斎藤泰一, 土井洋, 辻井重男：“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, Vol.44, No.8, pp.1913–1923, 2003.

[2] N.Ishida, S.Matsuo, and W.Ogata：“Divisible Voting Scheme,” Proc. ISC2003, LNCS 2851, pp.137–150, Springer, 2003.

[3] J.Camenisch and A.Lysyanskaya：“A signature scheme with efficient protocols,” Proc. SCN'02, LNCS 2576, pp.268–289, Springer, 2002.

[4] T.ElGamal：“A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms,” Proc. CRYPTO'84, LNCS 196, pp.10–18, Springer, 1985.

[5] Y.Desmedt and Y.Frankel：“Threshold cryptosystems,” Proc. CRYPTO'89, LNCS 196, pp.10–18, Springer, 1989.

[6] J.Camenisch and M.Michels：“Separability and efficiency for generic group signature schemes,” Proc. CRYPTO'99, LNCS 1666, pp.413–430, Springer, 1999.

[7] A.Chan, Y.Frankel, and Y.Tsiounis：“Easy Come - Easy Go Divisible Cash,” Proc. EUROCRYPT'98, LNCS 1403, pp.561–575, Springer, 1998.

付　　録

複数のリプリゼンテーションの知識を証明する SPK：

底 $g_1,\cdots,g_k$ に対する公開鍵 $y_1,\cdots,y_n$ のリプリゼンテーションの SPK

$$SPK\{(\alpha_{ij})_{i=1,\cdots,n;j\in\mathcal{J}_i} : \bigwedge_{i=1}^{n} y_i = \prod_{j\in\mathcal{J}_i} g_j^{\alpha_{ij}}\}(m)$$

は，すべての y_i のリプリゼンテーション α_{ij} $(i=1,\cdots,n;j\in\mathcal{J}_i)$ を知っているなら，次のように計算できる．まず $r_{ij}\in_R [0,2^{\ell_x+\tilde{\ell}})$ $(i=1,\cdots,n;j\in\mathcal{J}_i)$ を選び，$t_i := \prod_{j\in\mathcal{J}_i} g_j^{r_{ij}}$ $(i=1,\cdots,n)$ を計算する．次に以下の値を計算する．

$$c := \mathcal{H}(g_1||\cdots||g_k||y_1||\cdots||y_n||\mathcal{J}_1||\cdots||\mathcal{J}_n||t_1||\cdots||t_n||m)$$

$$s_{ij} := r_{ij} + c\alpha_{ij} \quad (i=1,\cdots,n;j\in\mathcal{J}_i)$$

このとき，$(c,(s_{ij})_{i=1,\cdots,n;j\in\mathcal{J}_i})\in\{0,1\}^{\ell_c}\times Z^{\sum_{i=1}^{n}|\mathcal{J}_i|}$ が署名となる．

一方，検証は

$$c = \mathcal{H}(g_1||\cdots||g_k||y_1||\cdots||y_n||\mathcal{J}_1||\cdots||\mathcal{J}_n \\ ||y_1^{-c}\prod_{j\in\mathcal{J}_1} g_j^{s_{1j}}||\cdots||y_n^{-c}\prod_{j\in\mathcal{J}_n} g_j^{s_{nj}}||m)$$

が成立するかどうかによって行われる．

リプリゼンテーションの成分の等しさを証明する SPK：

底 $g_1,\cdots,g_k$ に対する公開鍵 $y_1,\cdots,y_n$ のリプリゼンテーションの成分の一部が等しいことを証明する SPK

$$SPK\{(\alpha_1,\cdots,\alpha_u) : (y_1 = \prod_{j\in\mathcal{J}_1} g_j^{\alpha_{e_{1j}}}) \wedge\cdots\wedge \\ (y_n = \prod_{j\in\mathcal{J}_n} g_j^{\alpha_{e_{nj}}})\}(m)$$

は，対象の述語を満足する $(\alpha_1,\cdots,\alpha_u)$ を知っているなら，次のように計算できる．まず $i=1,\cdots,u$ について $r_i\in_R [0,2^{\ell_x+\tilde{\ell}})$ を選び，$t_i := \prod_{j\in\mathcal{J}_i} g_j^{r_{e_{ij}}}$ $(i=1,\cdots,n)$ を計算する．次に以下の値を計算する．

$$c := \mathcal{H}(g_1||\cdots||\mathcal{J}_n||\{e_{ij}\}_{i=1,\cdots,n;j\in\mathcal{J}_i}||t_1||\cdots||t_n||m)$$

$$s_i := r_i + c\alpha_i \quad (i=1,\cdots,u)$$

このとき，署名は $(c,s_1,\cdots,s_u)\in\{0,1\}^{\ell_c}\times Z^u$ となる．

一方，検証は

$$c = \mathcal{H}(g_1||\cdots||g_k||y_1||\cdots||y_n||\mathcal{J}_1||\cdots||\mathcal{J}_n|| \\ \{e_{ij}\}_{i=1,\cdots,n;j\in\mathcal{J}_i}||y_1^{-c}\prod_{j\in\mathcal{J}_1} g_j^{s_{e_{1j}}}||\cdots|| \\ y_n^{-c}\prod_{j\in\mathcal{J}_n} g_j^{s_{e_{nj}}}||m)$$

が成立するかどうかによって行われる．

いくつかの公開鍵のうちの一つのリプリゼンテーションの知識を証明する SPK：

底 $g_1,\cdots,g_k$ に対する公開鍵 $y_1,\cdots,y_n$ のうちの (少なくとも) 一つの y_i のリプリゼンテーションの知識を証明する SPK

$$SPK\{(\alpha_{ij})_{i=1,\cdots,n;j\in\mathcal{J}_i} : \bigvee_{i=1}^{n} y_i = \prod_{j\in\mathcal{J}_i} g_j^{\alpha_{ij}}\}(m)$$

は，例えば y_1 のリプリゼンテーションを知っているなら，次のように計算できる．まず $(r_{1j})_{j\in\mathcal{J}_1},(s_{ij})_{i=2,\cdots,n;j\in\mathcal{J}_i}\in_R [0,2^{\ell_x+\tilde{\ell}})$ と $c_2,\cdots,c_n\in_R\{0,1\}^{\ell_c}$ を選び，$t_1 := \prod_{j\in\mathcal{J}_1} g_j^{r_{1j}}$，$t_i := y_i^{-c_i}\prod_{j\in\mathcal{J}_i} g_j^{s_{ij}}$ $(i=2,\cdots,n)$ を計算する．次に以下の値を計算する．

$$c_1 := \mathcal{H}(g||\cdots||\mathcal{J}_n||t_1||\cdots||t_n||m)\oplus\bigoplus_{i=2}^{n} c_i$$

$$s_{1j} := r_{1j} + c_1\alpha_{1j} \quad (j\in\mathcal{J}_1)$$

このとき，署名は $(c_1,\cdots,c_n,(s_{ij})_{i=1,\cdots,n;j\in\mathcal{J}_i})\in(\{0,1\}^{\ell_c})^n\times Z^{\sum_{i=1}^{n}|\mathcal{J}_i|}$ となる．

一方，検証は

$$\bigoplus_{i=1}^{n} c_i = \mathcal{H}(g_1||\cdots||g_k||y_1||\cdots||y_n||\mathcal{J}_1||\cdots||\mathcal{J}_n \\ ||y_1^{-c_1}\prod_{j\in\mathcal{J}_1} g_j^{s_{1j}}||\cdots||y_n^{-c_n}\prod_{j\in\mathcal{J}_n} g_j^{s_{nj}}||m)$$

が成立するかどうかによって行われる．

リプリゼンテーションの一部がある範囲に存在することを証明する SPK：

底 $g_1,\cdots,g_k$ に対する公開鍵 y のリプリゼンテーションの一部が，ある整数 a と正整数 b に対して，範囲 $[a-2^{\tilde{\ell}}b,a+2^{\tilde{\ell}}b]$ に存在することを証明する SPK

$$SPK\{(\alpha_1,\cdots,\alpha_k) : y = \prod_{i=1}^{k} g_i^{\alpha_i} \\ \wedge\alpha_i\in[a-2^{\tilde{\ell}}b,a+2^{\tilde{\ell}}b]\}(m)$$

は，例えば $\alpha_1\in[a,a+b]$ であるような y のリプリゼンテーション $(\alpha_1,\cdots,\alpha_k)$ を知っているなら，次のように計算できる．まず $r_1\in_R[0,2^{\tilde{\ell}}b)$ と $r_2,\cdots,r_k\in_R[0,2^{\ell_x+\tilde{\ell}})$ を選び，$t := \prod_{i=1}^{k} g_i^{r_i}$ を計算する．次に以下の値を計算する．

$$c := \mathcal{H}(g_1||\cdots||g_k||y||t||m)$$

$$s_1 := r_1 + c(\alpha_1 - a)$$

$$s_i := r_i + c\alpha_i \qquad (i=2,\cdots,k)$$

もし $s_1\notin[cb,2^{\tilde{\ell}}b)$ であれば以上のことをやり直す．このとき，署名は $(c,s_1,\cdots,s_k)\in\{0,1\}^{\ell_c}\times[cb,2^{\tilde{\ell}}b)\times Z^{k-1}$ となる．

一方，検証は

$$c = \mathcal{H}(g_1||\cdots||g_k||y||y^{-c}g_1^{s_1+ca}\prod_{i=2}^{k} g_i^{s_i}||m) \ \wedge\ s_1\in[cb,2^{\tilde{\ell}}b)$$

が成立するかどうかによって行われる．