SCIS 2002 The 2002 Symposium on Cryptography and Information Security
Shirahama,Japan, Jan.29-Feb.1,2002
The Institute of Electronics,
Information and Communication Engineers

事前処理を仮定したMix-netの高速検証方式
A High-speed Verification Method for Mix-net Using Pre-computations

千田 浩司* 星野 文学*
Koji Chida Fumitaka Hoshino

あらまし 本稿ではネットワーク上で匿名を実現するMix-netについて，正当性が第三者に対しても検証可能な一方式を提案する．従来このような研究は，電子投票などのアプリケーションに利用可能なことから，様々な形で高速化が検討されているが，特に最近CRYPTO'01で古川，佐古により報告された方式は，証明が完全でないことが指摘されているものの，従来の$O(\kappa m n)$, $O(m n \log n)$に対して$O(m n)$で実現されることから注目を集めている（κ: セキュリティパラメータ，m: サーバ数，n: ユーザ数）．本稿で示す提案法は，選挙のように予めおおよそのユーザ数が与えられ，証明者と検証者が事前処理できる場合に特に有効であり，その場合事前処理を除いた実処理は証明検証合わせてべき乗演算約$11\,m n$となる．これは古川らの約$18\,m n$よりも効率が良い．また本稿では提案法の安全性がDecision Diffie-Hellman問題に帰着できることを証明し，更に提案法の処理演算に，ICICS'01で青木らにより報告された楕円暗号の高速化技法と，SCIS 2002で星野，阿部により報告された高速一括処理技法を用いることで，Mix-netを用いた電子投票がどれだけのユーザ数に対して適用できるか実装により確認する．

キーワード Mix-net, 全体検証，DDH問題，ゼロ知識証明

1 はじめに

1981年，Chaum[Cha81]により暗号技術を用いた匿名通信路を実現する方式*Mix-net*が提案された．基本的なMix-netは複数のサーバ（以降，これをMixサーバと呼ぶことにする）とそれらを直列につなぐネットワークからなり，複数の送信者が作成した暗号文のリストを入力とし，各Mixサーバは復号鍵を分散してもち，受信した暗号文をその復号鍵で復号するとともにランダムに置換して出力する．全サーバの入出力対応関係が知られない限り，Mix-netへの入力である暗号文のリストと最終的なMix-netからの出力である平文のリストとの対応がつかない．すなわち少なくとも一つのMixサーバが正しければ匿名性が満たされる．

その後Parkら[PIK94]により，Chaumの提案したRSA暗号ベースのMix-netをEl Gamal暗号ベースで実現する方式が提案された．これは従来と違い，Mixサーバは入力暗号文に対して，平文不変なまま再暗号化を施す．また従来のMixサーバの増加に伴い通信量・計算量ともに増加する点を解決しており効率が良い．更に1995年，佐古，Kilian[SK95]により，Parkらの方式に不正なMixサーバを検出できる機能を付加した方式が提案された．この報告以降，Parkらの方式をベースとして，Mixサーバに対する頑健性を備えた方式が様々な形で研究されている．特に尾形ら[OKS+97]，阿部[Abe99]の方式は，[SK95]とともにMixサーバの正当性が第三者に対しても検証可能であり，また入力暗号文を生成したユーザとMixサーバの結託による不正行為もできないような強い頑健性を備え持つ．このような性質を備えるMix-netは，*Universally Verifiable Mix-net*と呼ばれ，厳格な選挙などに有用であると考えられる．一方最近，古川，佐古[FS01]により，Universally Verifiable Mix-netの一実現方式が提案された．これは証明が完全でないことが指摘されているものの，[SK95],[OKS+97]の$O(\kappa m n)$, [Abe99]の$O(m n \log n)$に対して$O(m n)$で実現されることから注目を集めている（κ: セキュリティパラメータ，m: Mixサーバ数，n: ユーザ数）．

Universally Verifiable Mix-net実現の従来のアプローチは，Mixサーバの入出力が対応していることを決定づける，置換の存在を（非対話型）ゼロ知識証明によりMixサーバ自身が置換を明かすことなく証明することでゼロ知識性及び頑健性を満たしているが，本稿で示す提案法は，置換を予めゼロ知識証明によりコミットし，実処理ではMixサーバの入出力の対応を決定づける置換が，コ

* NTT情報流通プラットフォーム研究所 〒239-0847 神奈川県横須賀市光の丘1-1 NTT Information Sharing Platform Laboratories 1-1, Hikarinooka, Yokosuka-shi, Kanagawa-ken, 239-0847, Japan

ミットしたものと等しいことを計算量的な仮定の下で置換を明かすことなく証明することで処理の軽減を図っている．本方式は，選挙のように予めおおよそのユーザ数が与えられ，証明者と検証者が事前処理できる場合に特に有効であり，その場合事前処理を除いた実処理は，証明検証合わせてべき乗演算約 $11\,m\,n$ となる．これは古川らの約 $18\,m\,n$ よりも効率が良い．また提案法の安全性について，証明者に対する頑健性及び検証者に対するゼロ知識性がともに Decision Diffie-Hellman(DDH) 問題（[Bon98] 等を参照）に帰着できることを証明する．更に提案法の処理演算に，最近，青木ら及び星野，阿部によりそれぞれ報告された，楕円暗号の高速化技法 [AHK01] と，高速一括処理技法 [HA01] を用いた実装を行う．これにより，その処理量の大きさから 10,000 オーダの入力数が処理限度とされる Mix-net 利用の電子投票が，新たにどの程度まで適用範囲とできるかを確認する．

2 頑健な Mix-net

前節でも述べたように，Park らによる El Gamal 暗号ベースの Mix-net に対して頑健な方式がいくつか提案されている．本節では，その従来方式について簡単に紹介するが，そのためにまず [PIK94] の匿名性のみを満たした方式について触れておく．

q を safe prime，$< g >= \mathbf{G}_q, y \in \mathbf{G}_q$ を公開情報とする．但し $\mathbf{G}_q$ は位数 q の有限巡回群とし，g と y の間の離散対数を知るのは復号者のみとする（離散対数問題）．また本稿では今後これらの記号を継続して用いる．

はじめに n 人のユーザは自身のメッセージ m_i を El Gamal 暗号関数で暗号化した $(g^{w_i}, m_i y^{w_i})$ を公開する $(w_i \in_U \mathbf{Z}/q\mathbf{Z})$．次に各 Mix サーバの処理によりユーザと暗号文対の関係を切り離す．まず 1 番目の Mix サーバが n 個の暗号文対 $\{(g^{w_i}, m_i y^{w_i})\}$ を入力として，$\{1, \cdots, n\}$ から自身に写す置換関数全体の集合 Π から任意の要素を一つ選び（これを π とする），n 個の平文不変な再暗号文対 $\{(g^{w_{\pi(i)}+r_i}, m_{\pi(i)} y^{w_{\pi(i)}+r_i})\}$ を出力する $(r_i \in_U \mathbf{Z}/q\mathbf{Z})$．入出力の対応関係は DDH 問題に帰着される．同様の処理を k 番目 $(k = 2, \cdots, t)$ の Mix サーバが $k-1$ 番目の出力を入力として行う．これにより t 台全ての Mix サーバの秘密を知ることなく，n 個のユーザの入力暗号文と，t 番目の Mix サーバの出力との対応関係を知ることは DDH 問題に帰着される．

[PIK94] の方式に頑健性を兼ね備える方式として，Mix-net に応用することは特に述べられてないが，Cramer ら [CDS94] のゼロ知識証明のアイデアを利用することで，Mix サーバ数 m，ユーザ数 n に対して Mix サーバの処理の正当性証明及びその検証を $O(mn^2)$ で実現できることを先に述べておく．これに対して [SK95] では Cut-and-Choose 法を用いて $O(\kappa m\, n)$ で実現する方式を提案し (κ: セキュリティパラメータ)，[Abe99] では Permutation Network を利用した $O(m\, n \log n)$ での実現方式を与えている．[Abe99] は中小規模の匿名投票に適しており，現実的なユーザ数の範囲では [SK95] よりも効率が良い．一方最近，更に効率の良い方式が提案された [FS01]．これは入出力が 1 対 1 に対応していることを決定づける，置換行列の存在をゼロ知識で証明する方式であり，$O(mn)$ で実現できる[1]．[FS01] にもあるように，べき乗剰余演算の回数はそれぞれ [SK95] が $642n$，[Abe99] が $22(n \log n - n + 1)$，[FS01] が $18n + 18$ となる．

3 提案法

入出力が 1 対 1 に対応していることを決定づける置換の存在をゼロ知識で証明する従来のアプローチとは異なり，本稿で示す提案法は事前処理としてまず適当な n 個の値を Mix サーバに入力させ，その n 個の入力に対して，入出力の対応を隠蔽する変換処理を施した n 個の値を出力させ，その入出力が 1 対 1 に対応していることを決定づける，置換の存在をゼロ知識証明させておく．その後，実処理では同様の置換を用いることで，結果的に事前に証明した置換と等しいことをゼロ知識で証明する方式を与えており，検証者とおおよそのユーザ数が既知で事前処理が可能な状況であれば，事前処理を除いた実処理は従来法よりも効率が良い．

以降，Proof$[\{a_i\}\,;\, P(a_1, a_2, \cdots, a_r) = 1]$ で，$\{a_1, a_2, \cdots, a_r\}$ の部分集合 $\{a_i\}$ の要素を witness とし，また集合 $\{a_1, a_2, \cdots, a_r\} - \{a_i\}$ の要素と述語 P は公開としたゼロ知識証明を意味するものとする．その他，証明者（Mix サーバ）と検証者は事前に与えられているものとする[2]．

まず事前処理として証明者 $(\mathcal{P})$，検証者 $(\mathcal{V})$ 間で以下を行う．

[Protocol I]

$\mathcal{P}$(1): $z \in \mathbf{Z}/q\mathbf{Z}$ を生成した後，$G = g^z$ を計算し，G をコミットメントとして送る

$\mathcal{V}$(2): チャレンジ $\{h_i\} \in_U (\mathbf{G}_q)^n$ を送る

$\mathcal{P}$(3): $H_i = h_{\pi(i)}^z \quad (i = 1, 2, \cdots, n,\ \pi \in \Pi)$ を計算し，レスポンスとして H_i を送る

$\mathcal{PV}$(4): $\mathcal{P}$ は Proof$[\pi, z\,;\, H_i = h_{\pi(i)}^z \wedge G = g^z]$) を証明し，$\mathcal{V}$ はそれに対して検証を行う

$\mathcal{V}$(5): 最終的に Proof が正しければ受理し，そうでなければ拒絶する

[1] 但し現時点で証明が完全でないことが指摘されている

[2] 本方式は Mix サーバの正当性が第三者にも検証可能だが，その検証者が事前に立ち会うことで効率化を図ったものであることに注意．

$\mathcal{PV}(4)$ として例えば，[Abe99] を応用することで $O(n \log n)$ で実現できる．但し具体的なプロトコルは [Abe99] から明らかなため省略する．

次に実処理について説明する．まずはじめに，n 人のユーザが公開情報 g, y を用いて作成した，n 個の El Gamal暗号文対 $(G'_i, M'_i) \in (\mathbf{G}_q)^2 \quad i = 1,2,\cdots,n$ が公開された後，全Mixサーバが協力して，$\log_g \tilde{g} = \log_y \tilde{y}$ かつ離散対数 $\log_g \tilde{g}$ が未知であるような $\tilde{g}, \tilde{y} \in (\mathbf{G}_q)^2$ を作成する．これは全Mixサーバの秘密が知られることはなく，また，離散対数問題が困難であるという仮定の下で，各Mixサーバ j が，$\tilde{g}_j = g^{s_j}\ (s_j \in \mathbf{Z}/q\mathbf{Z}), \tilde{y}_j = y^{s_j}$ を計算，公開し，更に [CP93] の手法により Proof$[s_j\ ;\ \tilde{g}_j = g^{s_j} \wedge \tilde{y}_j = y^{s_j}]$ が証明されたうえで，$\tilde{g} = \prod_{j=1}^{t} \tilde{g}_j$, $\tilde{y} = \prod_{j=1}^{t} \tilde{y}_j$ とすることで実現できる．

以上の準備の下で，$(G_i, M_i) = (\tilde{g}G'_i, \tilde{y}M'_i)$ として１番目のMixサーバがまず以下を実行する．

1. $g, y, \{(G_i, M_i)\}$ を入力する
2. $\{r_i\} \in_U (\mathbf{Z}/q\mathbf{Z})^n$, $\pi^{-1} \in \Pi$ を選ぶ[3]
3. $\{(g^{r_i} G_{\pi^{-1}(i)}, y^{r_i} M_{\pi^{-1}(i)})\}$ を出力する

上記処理は [PIK94] と同様であり，$g, y, \{(G_i, M_i)\}$, $\{(g^{r_i} G_{\pi^{-1}(i)}, y^{r_i} M_{\pi^{-1}(i)})\}$ から π に関する情報を知ることはDDH問題に帰着される．

続いて，１番目のMixサーバに対して頑健性を備えるプロトコルを示す．ここで上記処理における１番目のMixサーバの出力を $\{(\tilde{G}_i, \tilde{M}_i)\}$ とする．

[Protocol II]

Input: $g, y, G, \{h_i\}, \{H_i\}, \{(G_i, M_i)\}, \{(\tilde{G}_i, \tilde{M}_i)\}$ $(i = 1,2,\cdots,n)$

$\mathcal{V}$**(1):** チャレンジ $a_0, \{a_i\} \in_U (\mathbf{Z}/q\mathbf{Z})^{n+1}$, $g', h' \in_U (\mathbf{G}_q)^2$ を送る

$\mathcal{P}$**(2):** $X = h'^{s} (\prod h_i^{a_i})^z\ (s \in_U \mathbf{Z}/q\mathbf{Z})$,
$Y = g'^{x} (g y^{a_0})^{\sum a_i r_i}\ (x \in_U \mathbf{Z}/q\mathbf{Z})$ を計算し，レスポンスとして X, Y を送る

$\mathcal{PV}$**(3):** $Z = \prod (\tilde{G}_i \tilde{M}_i^{a_0})^{a_i}$, $W = Z/Y$ として，$\mathcal{P}$ は
Proof$[s, z, x, \pi, \sum a_i r_i\ ;$
$\quad X = h'^{s} (\prod h_i^{a_i})^z$
$\wedge\ X = h'^{s} \prod H_i^{a_{\pi(i)}}$
$\wedge\ G = g^z$
$\wedge\ Y = g'^{x} (g y^{a_0})^{\sum a_i r_i}$
$\wedge\ W = (1/g')^x \prod (G_i M_i^{a_0})^{a_{\pi(i)}}]$
を証明し[4]，$\mathcal{V}$ はそれに対して検証を行う

$\mathcal{V}$**(4):** 最終的にProofが正しければ受理し，そうでなければ拒絶する

上記 $\mathcal{PV}(3), \mathcal{V}(4)$ の処理は具体的に以下のように行う．

[Protocol II-i]

$\mathcal{P}$**(3-1):** $X_0 = h'^{s_0} (\prod h_i^{a_i})^{z_0}\ (s_0, z_0 \in_U (\mathbf{Z}/q\mathbf{Z})^2)$,
$X_1 = h'^{s_0} \prod H_i^{w_i}\ (\{w_i\} \in_U (\mathbf{Z}/q\mathbf{Z})^n)$,
$G_0 = g^{z_0}$,
$Y_0 = g'^{x_0} (g y^{a_0})^{d_0}\ (x_0, d_0 \in_U (\mathbf{Z}/q\mathbf{Z})^2)$,
$W_0 = (1/g')^{x_0} \prod (G_i M_i^{a_0})^{w_i}$ をコミットメントとして送る

$\mathcal{V}$**(3-2):** チャレンジ $c \in_U \mathbf{Z}/q\mathbf{Z}$ を送る

$\mathcal{P}$**(3-3):** $s_0 - cs, \{w_i - c a_{\pi(i)}\}, z_0 - cz, x_0 - cx, d_0 - c \sum a_i r_i$ をレスポンスとして送る

$\mathcal{V}$**(4):** $h'^{s_0 - cs} (\prod h_i^{a_i})^{z_0 - cz} = X_0 / X^c$
$\wedge\ h'^{s_0 - cs} \prod H_i^{w_i - c a_{\pi(i)}} = X_1 / X^c$
$\wedge\ g^{z_0 - cz} = G_0 / G^c$
$\wedge\ g'^{x_0 - cx} (g y^{a_0})^{d_0 - c \sum a_i r_i} = Y_0 / Y^c$
$\wedge\ (1/g')^{x_0 - cx} \prod (G_i M_i^{a_0})^{w_i - c a_{\pi(i)}} = W_0 / W^c$
が成り立てば受理し，そうでなければ拒絶する

以下同様の処理をMixサーバ j と検証者について行う $(j = 2,\cdots,t)$．ここで２番目のMixサーバは，$\{(\tilde{G}_i, \tilde{M}_i)\}$ を $\{(G_i, M_i)\}$ と置き換えて入力とすれば良い．続けて３番目以降も同様である．また，安全性については次節で説明する．

4 考察

ここでは前節で示した提案法が安全である，すなわち，

- Mixサーバの入出力に対する正当性証明のリストからは，入出力の対応関係に関する情報を与えない（ゼロ知識性）
- Mixサーバが不正な処理を行った場合には，その事実が明らかになる（頑健性）

を提案法は満たすことを示す．

はじめに Protocol II-i の安全性について考察する．

Lemma 1 Protocol II-i は *honest verifier zero-knowledge* である．

Proof 証明者，検証者間で正しく行われた Protocol II-i のリストを $\mathcal{R}$，シミュレータを $\mathcal{S}$ とする．このとき $\mathcal{S}$ にリスト $\mathcal{R} = \{(X, G, Y, W), (X_0, X_1, G_0, Y_0, W_0), c, (s_0 - cs, \{w_i - c a_{\pi(i)}\}, z_0 - cz, x_0 - cx, d_0 - c \sum a_i r_i)\}$ を入力した場合，$\mathcal{S}$ の動作として $s', \{w'_i\}, z', x', d' \in_U (\mathbf{Z}/q\mathbf{Z})^{n+4}$ を選んだ後，$X'_0 = X^c h'^{s'} (\prod h_i^{a_i})^{z'}$, $X'_1 = X^c h'^{s'} \prod H_i^{w'_i}$, $G'_0 = G^c g^{z'}$, $Y'_0 = Y^c g'^{x'} (g y^{a_0})^{d'}$, $W'_0 =$

[3] ここで π^{-1} は Protocol I で選んだ置換 π の逆関数

[4] ここで X について２通りの表現でゼロ知識証明を行う．すなわち，$h', \prod h_i^{a_i}$ を基底としたときの離散対数と，$h', \{H_i\}$ を基底としたときの離散対数を知っていることをゼロ知識で証明する．

$W^c(1/g')^{x'}\prod(G_iM_i^{a_0})^{w_i'}$ を計算して $\mathcal{F}=\{(X,G,Y,W),(X_0',X_1',G_0',Y_0',W_0'),c,(s',\{w_i'\},z',x',d')\}$ を出力する．明らかにリスト $\mathcal{F}$ は Protocol II-i を受理し，かつ $\mathcal{F}$ の取り決め方から，$\mathcal{R}$ と $\mathcal{F}$ は完全識別不可能である．

□

Lemma 2　Protocol II-i は健全性を満たす．

Lemma 2 について，Protocol II-i を rewind したとき，$\mathcal{V}$ が witness $(s, z, x, \pi, \sum a_ir_i)$ を抽出できることは明らか．

次に提案方式全体の安全性について議論する．そのために以下の補題と安全性のモデルをまず定義する．

Lemma 3　Protocol II におけるレスポンス X,Y は witness $(s,z,x,\pi,\sum a_ir_i)$ に対してゼロ知識である．

Proof　$X=h'^s(\prod h_i^{a_i})^z$ について，乱数 $s\in\mathbf{Z}/q\mathbf{Z}$ は X にのみ用いられ，かつ z とは独立である．すなわち X は s によってランダマイズされるためゼロ知識である．同様に，$Y=g'^x(gy^{a_0})^{\sum a_ir_i}$ について，乱数 $x\in\mathbf{Z}/q\mathbf{Z}$ は Y にのみ用いられ，かつ $\sum a_ir_i$ とは独立である．すなわち Y は x によってランダマイズされるためゼロ知識である．

□

Definition 1　攻撃者 $\mathcal{A}$ はあるセキュリティパラメータ ρ に対して多項式限定で動作し，$t-1$ 台の Mix サーバ，$n-2$ 人のユーザを自由に操作できるものとし，操作できない唯一の Mix サーバの入出力を対応づける置換を求めることを目的とする．

題意より一般性を失うことなく，攻撃者 $\mathcal{A}$ は少なくとも操作できない 2 入力の El Gamal 暗号文対 (G_1,M_1), (G_2,M_2) と操作できない 1 番目の Mix サーバの 2 出力 $(\tilde{G}_1,\tilde{M}_1),(\tilde{G}_2,\tilde{M}_2)$ との対応関係を求めることが必要となる．但し便宜上，以降上記の入出力をそれぞれ $\{(G_0,M_0),(G_1,M_1)\}$, $\{(\tilde{G}_b,\tilde{M}_b),(\tilde{G}_{\hat{b}},\tilde{M}_{\hat{b}})\}$　$b\in\{0,1\}$, $\hat{b}=b\oplus 1$ と置き換える．

Definition 2　q を safe prime, g を位数 q の有限巡回群を生成する適当な元とする．このとき入力 $\mathcal{I}_0=(g,g^\alpha,g^\beta,g^\gamma)$ が与えられ，

$$\begin{cases}\mathcal{I}\in\mathbf{D} & \text{if}\quad \alpha,\beta\in(\mathbf{Z}/q\mathbf{Z})^2,\gamma=\alpha\beta\\ \mathcal{I}\in\mathbf{R} & \text{if}\quad \alpha,\beta,\gamma\in(\mathbf{Z}/q\mathbf{Z})^3\end{cases}$$

としたとき，$\mathcal{I}$ が集合 $\mathbf{D},\mathbf{R}$ のどちらに属するか，あるセキュリティパラメータ ρ に対して多項式時間で 1/2 以上の無視できない確率で求めることは困難である．

Definition 2 は Decision Diffie-Hellman (DDH) 仮定と呼ばれる計算量的困難性に基づいた DDH 問題に対する仮定である (厳密な定義については [Bon98] 等を参照)．

上記準備の下で，本題である提案法全体の安全性について以下の定理が成り立つ．

Theorem 1　$\mathcal{P},\mathcal{V}$ を正しいプレイヤーとし，攻撃者を Definition 1 で定義した $\mathcal{A}$ とする．このとき Protocol I, II から，攻撃者 $\mathcal{A}$ の目的を無視できない確率で達成させるオラクルが存在する[5] ならば，それを用いることにより Definition 2 で定義した仮定に反して，DDH 問題を解くようなオラクルを構成することができる．

Theorem 2　$\mathcal{V}$ が Protocol I, II を正しく受理するならば，離散対数問題が困難である仮定の下で，ある $\pi\in\Pi$, $\{c_i\}\in(\mathbf{Z}/q\mathbf{Z})^n$ が存在し，$\mathcal{P}$ のコミットメント $G=g^z$ に対して $H_i=h_{\pi(i)}^z$, $(\tilde{G}_i,\tilde{M}_i)=(g^{c_i}G_{\pi^{-1}(i)},y^{c_i}M_{\pi^{-1}(i)})$ が成り立つ．

Proof (Theorem 1)　Protocol I の $\mathcal{P}(3)$ に表れる Proof がゼロ知識で構成できることは [Abe99] から明らかなため省略する．また，Protocol II-i に表れる Proof と，Protocol II に表れる X,Y がゼロ知識であることは Lemma 1 及び Lemma 3 により既に示されている．すなわち Definition 1 直後の段落の記述と，Protocol I, II から，$\mathcal{I}_1=(g,y,g^z,h_0,h_1,h_{\hat{b}}^z,h_{\hat{\hat{b}}}^z,(G_0,M_0),(G_1,M_1),(\tilde{G}_b,\tilde{M}_b),(\tilde{G}_{\hat{b}},\tilde{M}_{\hat{b}}))$ を入力として，多項式時間で 1/2 以上の無視できない確率で $\mathcal{O}_1=b$ を出力できるオラクル $\mathcal{B}$ を用いて DDH 問題を解くオラクル $\mathcal{D}$ が構成できることを示せば良い．

Definition 2 で示した DDH 問題のインスタンス $\mathcal{I}_0=(g,g^\alpha,g^\beta,g^\gamma)$ を入力として，以下のような動作をする $\mathcal{D}$ を構成する．

1. $\mathcal{I}_0'=(g,g^\alpha,g^{\beta v_0'+v_1'},g^{\gamma v_0'+\alpha v_1'})$ を計算する $(v_0',v_1'\in_U(\mathbf{Z}/q\mathbf{Z})^2)$
2. $\mathcal{I}_0'$ から $\mathcal{I}_1'=(g,g^\alpha,g^{\alpha t},g^{\beta v_0 t},g^{\beta v_1 t},g^{\gamma v_b t},g^{\gamma v_{\hat{b}} t},(g^\beta,m_0g^\gamma),(g^{\beta v},m_1g^{\gamma v}),(g^{\beta v^b+v_b'},m_bg^{\gamma v^b+\alpha v_b'}),(g^{\beta v^{\hat{b}}+v_{\hat{b}}'},m_{\hat{b}}g^{\gamma v^{\hat{b}}+\alpha v_{\hat{b}}'}))$ を計算する $(t,v_0,v_1,v\in_U(\mathbf{Z}/q\mathbf{Z})^4,m_0,m_1\in_U(\mathbf{G}_q)^2,b\in_U\{0,1\},\hat{b}=b\oplus 1)$
3. $\mathcal{I}_1'$ を $\mathcal{B}$ に入力し，出力 $\mathcal{O}_1'\in\{0,1\}$ を受け取る
4. $\mathcal{O}_1'=b$ ならば $\mathcal{I}\in\mathbf{D}$ と判断して“ 1 ”を出力し，$\mathcal{O}_1'\neq b$ ならば $\mathcal{I}\in\mathbf{R}$ と判断して“ 0 ”を出力する

[5] このオラクルの動作として，もし分布の異なるインスタンスが入力された場合は，出力値域からランダムな値を選んで多項式時間内でそれを出力するものとする．

ここで $\mathcal{I}_0'$ について，$\mathcal{I}_0 \in \mathbf{D}$ すなわち $\gamma = \alpha\beta$ のとき，$\mathcal{I}_0' = (g, g^{\alpha}, g^{\beta v_0' + v_1'}, g^{\alpha(\beta v_0' + v_1')})$ となり，$\mathcal{I}_0$ と $\mathcal{I}_0'$ の分布は一致する．一方 $\mathcal{I}_0 \in \mathbf{R}$ すなわち $\gamma \in_U \mathbf{Z}/q\mathbf{Z}$ のとき，$\mathcal{I}_0' = (g, g^{\alpha}, g^{\beta v_0' + v_1'}, g^{\gamma v_0' + \alpha v_1'})$ となり，$\beta v_0' + v_1'$ と $\gamma v_0' + \alpha v_1'$ は独立であるから，やはり $\mathcal{I}_0$ と $\mathcal{I}_0'$ の分布は一致する．また，$\mathcal{I}_0$ と $\mathcal{I}_0'$ の分布が一致し，かつ $\mathcal{I}_0 \in \mathbf{D}$ であれば $\mathcal{I}_1$ と $\mathcal{I}_1'$ の分布が一致することは明らか．逆に $\mathcal{I}_0 \in \mathbf{R}$ であれば，圧倒的確率で $\mathcal{I}_1$ と $\mathcal{I}_1'$ の分布は一致しない．

上記に基づいて，オラクル $\mathcal{D}$ を用いたときの DDH 問題の攻撃に対する成功確率について述べる．まず $\mathcal{B}$ の出力は $\mathcal{I}_0, \mathcal{I}_1'$ に依存することが分かり，$\mathcal{I}_0 \in \mathbf{D}$ のとき $\mathcal{B}$ は多項式時間で $\mathcal{O}_1' = b$ なる出力 $\mathcal{O}_1'$ を確率 $1/2 + \theta$ で返す．ただし θ はあるセキュリティパラメータ κ に対して $\theta > 1/2^{\kappa}$ を満たす値とする．逆に $\mathcal{I}_0 \in \mathbf{R}$ のとき $\mathcal{B}$ は多項式時間で $b = \mathcal{O}_1'$ なる出力 $\mathcal{O}_1'$ を確率 $1/2 + \epsilon$ で返す．ただし ϵ はあるセキュリティパラメータ κ に対して $|\epsilon| < 1/2^{\kappa}$ を満たす値とする．これは Theorem 1 の注釈にもあるように，分布が一致しないインスタンスを $\mathcal{B}$ に入力した場合，$\mathcal{B}$ は多項式時間内で $\mathcal{O}_1' \in_U \{0,1\}$ を出力することによる．すなわちオラクル $\mathcal{D}$ が DDH 問題の攻撃に成功する確率はおよそ $1/2(1/2+\theta) + 1/2(1/2+\epsilon) = 1/2 + (\theta+\epsilon)/2$ となり無視できない．

□

Proof (Theorem 2)　Lemma 2 から，Protocol II の $\mathcal{P}(2)$ は健全性を満たすことが分かる．この Lemma 2 の結果と g', h' は他のどの元とも無関係に一様に取られていることから，$\mathcal{V}$ が $\mathcal{PV}(3)$ の Proof を $\mathcal{V}(4)$ で受理するために離散対数問題が困難である仮定の下で $\mathcal{P}$ は

$$X' = (\prod h_i^{a_i})^z \wedge\ X' = \prod H_i^{u_i}$$
$$\wedge\, G = g^z \wedge\ Y' = (gy^{a_0})^e \wedge\ W' = \prod (G_i M_i^{a_0})^{u_i} \quad (1)$$

を満たす離散対数 $\{u_i\}, e \in (\mathbf{Z}/q\mathbf{Z})^{n+1}$ を知っている必要がある $(X' = X/h'^s, Y' = Y/g'^x, W' = Z/Y')$．はじめに $\{u_i\}$ については，(1) 及び Protocol I より

$$(\prod h_i^{a_i})^z = \prod (h_{\pi(i)}^z)^{u_i}$$

であるから，$\{h_i\}$ はそれぞれ一様に取られているため離散対数問題が困難である仮定の下で $u_i = a_{\pi(i)}$ と定まる．すなわち (1) について

$$Y' = (gy^{a_0})^e \wedge\ W' = \prod (G_i M_i^{a_0})^{a_\pi(i)} \quad (2)$$

を満たす離散対数 e を $\mathcal{P}$ は知っている必要がある．いま，$W' = Z/Y' = \prod (\tilde{G}_i \tilde{M}_i{}^{a_0})^{a_i} / (gy^{a_0})^e$ であるから，(2) より

$$\prod (\tilde{G}_i \tilde{M}_i{}^{a_0})^{a_i} = (gy^{a_0})^e \prod (G_i M_i^{a_0})^{a_\pi(i)} \quad (3)$$

を得る．一方，題意から $\mathcal{P}$ は任意の $a_0, \{a_i\}$ に対して，$gy^{a_0}, \{G_i M_i^{a_0}\}$ を基底とした (3) の左辺 $\prod (\tilde{G}_i \tilde{M}_i{}^{a_0})^{a_i}$ の離散対数を知っている必要がある．そのため，g , y , $\{G_i\}$, $\{M_i\}$ の間の離散対数は未知なことより[6]，離散対数問題が困難である仮定の下で

$$\begin{cases} \tilde{G}_i = g^{c_i} \prod_{j=1}^n G_j^{\sigma_{i,j}} \ \{c_i\}, \{\sigma_{i,j}\} \in (\mathbf{Z}/q\mathbf{Z})^{n^2+n} \\ \tilde{M}_i = y^{d_i} \prod_{j=1}^n M_j^{\tau_{i,j}} \ \{d_i\}, \{\tau_{i,j}\} \in (\mathbf{Z}/q\mathbf{Z})^{n^2+n} \end{cases} \quad (4)$$

を満たす $\{c_i\}, \{d_i\}, \{\sigma_{i,j}\}, \{\tau_{i,j}\}$ から $\mathcal{P}$ は $\{\tilde{G}_i\}, \{\tilde{M}_i\}$ を構成する必要がある．すると (3),(4) から

$$\sum a_i c_i = \sum a_i d_i = e$$

であるから，任意の $\{a_i\}$ についてこれが成り立つために

$$c_i = d_i \quad (i = 1, 2, \cdots, n) \quad (5)$$

を得る．つまり (3) は (4),(5) より

$$\prod_{i=1}^n (\prod_{j=1}^n G_j^{\sigma_{i,j}} M_j^{\tau_{i,j} a_0})^{a_i} = \prod_{i=1}^n (G_i M_i^{a_0})^{a_\pi(i)} \quad (6)$$

とできるが，任意の $\{a_i\}$ について (6) が成り立つために

$$\begin{cases} \sigma_{i,j} = \tau_{i,j} = 1 \quad (\, j = \pi^{-1}(i)\,) \\ \sigma_{i,j} = \tau_{i,j} = 0 \quad (\text{otherwise}) \end{cases} \quad (7)$$

が必要十分となる．最終的に (4) に (5),(7) の結果を代入することで $(\tilde{G}_i, \tilde{M}_i) = (g^{c_i} G_{\pi^{-1}(i)}, y^{c_i} M_{\pi^{-1}(i)})$ を得る．

□

5　実装

本節では，提案法による Universally Verifiable Mix-net が実時間でどの程度の入力数を処理できるか実装により確認する．なお処理演算として，最近報告された楕円暗号の高速化技法 [AHK01] と，高速一括処理技法 [HA01] を用いた実装を行う．ここでの実装目的は，1 節でも触れたように，その処理量の大きさから 10,000 オーダの入力数が処理限度とされる Mix-net 利用の電子投票が，新たにどの程度まで適用範囲とできるかを確認するためであるため，従来法との実装による比較は特に行わない．但し従来法の理論的な計算量は 2 節で取り上げている．

下表のうち，“ 暗号処理 ”の項目で，[PIK94] に示された El Gamal 暗号の再暗号化とリストの置換に要する処理時間を意味する．また“ 正当性証明 ”，“ 検証 ”の項目で，それぞれ証明者が証明リスト作成に要する処理時間，検証者がそれを検証するのに要する処理時間を意味する．また今回の実装は Protocol II について非対話型

[6] 一般にはユーザと Mix サーバの結託によりこれはいえないが，$\{(G_i, M_i)\}$ はユーザの暗号文対 $\{(G_i', M_i')\}$ にそれぞれ $\tilde{g}, \tilde{y}$ を掛け合わせたものであることから，この議論が成り立つ．

ゼロ知識証明を行い，また以下の処理時間を無視している．

- Protocol I で示した，Mix サーバの事前処理及びその検証処理
- Mix サーバによる $\tilde{g}, \tilde{y}$ の作成と正当性証明処理及びその検証処理
- Protocol II に表れるチャレンジを生成するハッシュ処理 (実装では乱数を生成)
- データ入出力の遅延 (実装結果は全て on メモリ)

表： 提案法の処理時間（処理単位：秒）

入力数	暗号処理	正当性証明	検証
10^3	0.178	0.525	0.524
10^4	1.186	5.255	5.244
10^5	8.788	51.897	51.784
10^6	72.228	520.903	519.772

(計算環境)
ハード：COMPAQ 21264 500MHz
メモリ：512MB
ソフトウェア (OS)：OSF1 V4.0 1091 alpha
言語：DEC C V5.8-011
使用した群：OEF([AHK01]), $2^{61}-1$ の 3 次拡大,
(位数 183bit)

6 まとめ

ネットワーク上で匿名を実現する Mix-net について，正当性が第三者に対しても検証可能な一方式を提案した．今回示した提案法は，選挙のように予めおおよそのユーザ数が与えられ，証明者と検証者が事前処理できる場合に特に有効であり，その場合事前処理を除いた実処理は証明検証合わせてべき乗演算約 11 mn で済む（m: Mix サーバ数，n: ユーザ数）．これは従来最も効率が良いとされた約 18 mn よりも更に効率が良い．また提案法の安全性が Decision Diffie-Hellman 問題に帰着できることを証明したうえで，提案法の有効範囲を実装により確認した．具体的には，提案法の処理演算に既存の楕円暗号の高速化技法と高速一括処理技法を用いて実装した結果，従来 10,000 オーダの入力数が処理限度とされる Mix-net を用いた電子投票を，1,000,000 オーダに拡張できることを示した．

謝辞

有益なアドバイスを頂いた，NTT 情報流通プラットフォーム研究所 齊藤 泰一 氏，阿部 正幸 氏に感謝致します．

参考文献

[Abe99] M. Abe, “Mix-networks on permutation networks,” Proc. of ASIACRYPT’99, LNCS 1716, Springer-Verlag, pp.258-273, 1999

[AHK01] K. Aoki, F. Hoshino, and T. Kobayashi, “A Cyclic Window Algorithm for ECC Defined over Extension Fields,” Proc. of ICICS 2001, LNCS 2229, Springer-Verlag, pp.62-73, 2001

[Bon98] D. Boneh, “The Decision Diffie – Hellman Problem” Proc. of ANTS – III, LNCS 1423, Springer-Verlag, pp.48-63, 1998

[Cha81] D. L. Chaum, “Untraceable electronic mail, return address, and digital pseudonyms,” Communications of the ACM, 24:84-88, 1981

[CP93] D. L. Chaum and T. P. Pedersen, “Wallet Databases with Observers,” Proc. of CRYPTO’92, LNCS 740, Springer-Verlag, pp.89-105, 1993

[CDS94] R. Cramer, I. Damgård, and B. Schoenmakers, “Proofs of Partial Knowledge and Simplified Design of Witness Hiding Protocols,” Proc. of CRYPTO’94, LNCS 839, Springer-Verlag, pp.174-187, 1994

[FS01] J. Furukawa and K. Sako, “An Efficient Scheme for Proving a Shuffle,” Proc. of CRYPTO 2001, LNCS 2139, Springer-Verlag, pp.368-387, 2001

[HA01] 星野 文学，阿部 正幸，「Batch 検証と大規模冪乗アルゴリズムについて」SCIS 2002, 2002 年 1 月

[OKS+97] W. Ogata, K. Kurosawa, K. Sako, and K. Takatani, “Fault tolerant anonymous channel,” Proc. of ICICS’98, LNCS 1334, Springer-Verlag, pp.440-447, 1997

[PIK94] C. Park, K. Itoh, and K. Kurosawa, “Efficient anonymous channel and all/nothin election scheme,” Proc. of EUROCRYPT’93, LNCS 765, Springer-Verlag, pp.248-259, 1994

[SK95] K. Sako and J. Kilian, “Receipt-free mix-type voting scheme – a practical solution to the implementation of a voting booth –,” Proc. of EUROCRYPT ’95, LNCS 921, Springer-Verlag, pp.393-403, 1995