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

楕円暗号による MIX-net を用いた実用的な電子投票システム

Electronic Voting System with MIX-net using Elliptic Curve Cryptosystem

鬼頭宏幸* 星野文学* 千田浩司*
Hiroyuki Kito Fumitaka Hoshino Koji Chida

あらまし
電子投票システムでは、処理の正当性と匿名性の両立が必要になるため、全体検証可能な MIX-net が有用な技術となる。しかし一般に全体検証可能 MIX-net は処理が重く、実用化が困難であった。
そこで本稿では、入力暗号文に依存しない べき乗演算を開票開始前に行ない、さらに MIX サーバの処理を並列化する方式を提案する。これにより、MIX サーバの台数によらず、開票開始から $2(NlogN-N+1)$ 回のべき乗演算の時間で、全 MIX サーバの証明生成処理を完了することができる。また、OEF を用いた楕円暗号実装技術による実装評価も行なう。

キーワード 電子投票, 匿名通信路, MIX-net, 楕円暗号, OEF

1 はじめに

知事選挙や市長選挙、都道府県議員選挙などで電子投票システムの利用を認める「地方自治体電子投票特例法案」が近く成立する見通しになり、電子投票システムが注目されている。電子投票システムでは、従来の紙を使った投票とは異なり、開票処理の電子化により開票操作がブラックボックス化されるため、「開票処理中に、票の水増し、すり替え等の不正があったのではないか」という投票者の疑念を払うことが必要となる。そのため、処理の正当性検証と匿名性が両立できる全体検証可能 MIX-net が有用な技術となる。

MIX-net は、複数の MIX サーバから構成され、暗号文の列を入力とし、別の暗号文に変換 (もしくは入力暗号文を復号) し、順序を入れ替えて出力する (以降、ランダム化置換と呼ぶ)。さらに全体検証可能 MIX-net では、任意の第三者が入力の列と出力の列が一対一に対応することを検証できる。これを電子投票に適用すると、暗号化された投票内容は MIX-net に入力され、MIX-net によって別の暗号文に変換し順序を入れ替えて出力され、この出力暗号文が復号される。これにより、投票者と投票内容の関係が分らなくなり、投票の秘密を確保することができる。また、任意の第三者 (紙の投票における開票立会人に相当) は、開票処理が電子化されているにも拘わらず、処理の正当性を確認することができる。

* 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 239-0847, JAPAN

阿部らは、N 個の暗号文が入力された時、1 つの MIX サーバ毎に、$12(NlogN-N+1)$ 回のべき乗演算で処理できる効率的な全体検証可能 MIX-net 方式を提案している [1][2]。この方式では、各 MIX サーバにおいて、ランダム化置換処理及び、任意の第三者が正当性検証するのに必要な付加情報を生成する処理 (以降、証明生成と呼ぶ) を行なう。しかし、N 個の入力が与えられてから、複数の MIX サーバが 1 台ずつ順番にランダム化置換処理、証明生成を行なうため、開票開始から復号開始まで、M 台の MIX サーバで、$M \times 12(NlogN-N+1)$ 回のべき乗演算の時間が必要となる。

そこで本稿では、上記方式をベースに入力暗号文に依存しない べき乗演算を開票開始前に処理することで、開票開始後のべき乗演算数の削減を行なう。さらに、本提案方式では、MIX サーバがランダム化処理を完了し、証明生成を処理している間に、次段 MIX サーバのランダム化置換を並列して処理を行なう。これにより、MIX サーバの台数によらず、開票開始から $2(NlogN-N+1)$ 回のべき乗演算の時間で、全 MIX サーバが証明生成処理を完了することができる。

一方、青木、小林らは、楕円暗号を最適拡大体 (OEF; Optimal Extention Field) を用いて高速にソフトウェア実装する技術について提案している [3]。本稿では、この実装技術を用いた電子投票システムの実装評価についても述べる。

2　準備

2.1　登場エンティティ

本稿での登場エンティティは、図 1 に挙げるように、N 人の投票者と M 台の MIX サーバと復号者からなる。投票者は、投票内容を暗号化して、MIX サーバに送信する。投票締切り後、MIX サーバは投票データをランダム化置換し、その結果を復号者が復号する。なお復号者は、復号鍵を秘密分散させ複数人で復号しても良い。検証者 (任意の第三者) は、各 MIX サーバの入出力の列が一対一に対応することを検証する。

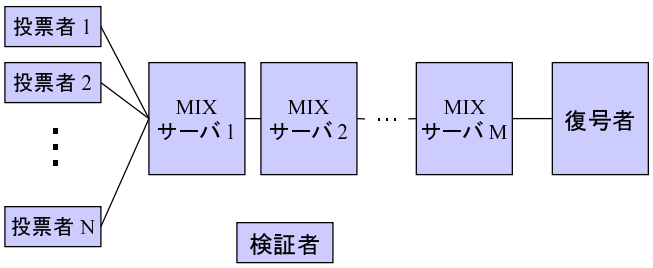

図 1: 登場エンティティ

2.2　システム要件

現実のシステムにおいては、票のすり替え等の MIX サーバの不正行為によって、MIX-net から不正な結果が出力される可能性は低い。これは、MIX サーバが不正行為を行なったとしても、任意の第三者による正当性検証時に不正が発覚し、同 MIX サーバに対して何らかの処罰が下されることになり、不正 MIX サーバにとって利益がないためである。

よって、開票後に MIX サーバの正当性検証ができれば十分であり、本稿では開票開始から開票結果を出力するまでの処理の効率化に重点を置く。

2.2.1　全体検証可能性

任意の第三者は MIX-net における入出力暗号文のすり替え等の不正行為を検出することができる。

2.2.2　匿名性

投票者と投票内容との対応 (MIX-net への入出力の対応) を知ることができない。

3　従来方式

本節では、本稿のベースとなる阿部らの方式 [1][2] について述べる。本方式では、ゼロ知識証明で入出力対応の存在を証明できる 2 入力 2 出力の置換器を基本単位 (以降セルと呼ぶ) とし、それを組み合わせることによって、2^k 入力 2^k 出力 (k は整数) の全体検証可能な MIX-net を実現する。

MIX-net は複数の MIX サーバから構成され、各 MIX サーバは、平文 m、生成元 g、公開鍵 y に対する ElGamal 暗号文 $(M,G) = (my^t, g^t)$ (t は乱数) を入力とし、秘密の乱数 r を用いて $(M',G') = (My^r, Gg^r)$ を出力する。このとき、(M,G) に対して (M',G') は、同じ平文 m、同じ生成元 g、同じ公開鍵 y に対する異なる乱数 $t' = t + r$ による ElGamal 暗号文となっている。この処理をランダム化という。

3.1　2 入力 2 出力セルの処理

各セルでは、2 つの入力 ElGamal 暗号文

$$\begin{aligned}(M_0,G_0) &= (m_0 y^{t_0}, g^{t_0})\\ (M_1,G_1) &= (m_1 y^{t_1}, g^{t_1})\end{aligned}$$

をランダム化し、さらに置換パラメータを用いて順序の入れ替えを行ない、

$$\begin{aligned}(M'_0,G'_0) &= (m'_0 y^{t'_0}, g^{t'_0})\\ (M'_1,G'_1) &= (m'_1 y^{t'_1}, g^{t'_1})\end{aligned}$$

を出力する。また、

$$(m_0 = m'_0 \vee m_1 = m'_1) \wedge (m_0 = m'_1 \vee m_1 = m'_0)$$

をゼロ知識証明で示すことで入出力対応が存在することを示す。任意の第三者はこの処理で生成された付加情報を用いて正当性の検証を行なう。さらに星野らは本処理を

$$(m_0 = m'_0 \vee m_0 = m'_1) \wedge (m_0 m_1 = m'_0 m'_1)$$

とすることで効率化できることを提案している [2]。

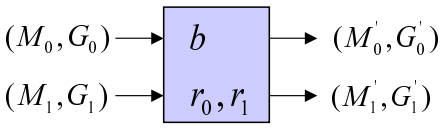

図 2: 置換及びランダム化

3.1.1　ランダム化置換処理

以降、置換パラメータを $b \in \{1,0\}$ とし、$\bar{b} = b \oplus 1$ とする。

入力 : M_0, G_0, M_1, G_1, y, g

出力 : $M'_0, G'_0, M'_1, G'_1, r_0, r_1$

べき乗演算数 : 4 回

処理内容 :

1. 乱数 r_0, r_1 を生成する。

2. 以下を計算する。

$$\begin{aligned}(M'_b, G'_b) &= (M_0 y^{r_0}, G_0 g^{r_0}) \\ &= (m_0 y^{t_0+r_0}, g^{t_0+r_0}) \\ (M'_{\bar{b}}, G'_{\bar{b}}) &= (M_1 y^{r_1}, G_1 g^{r_1}) \\ &= (m_1 y^{t_1+r_1}, g^{t_1+r_1})\end{aligned}$$

3.1.2 証明生成処理

以降、h は公開されたハッシュ関数とする。

$(m_0 = m'_0 \vee m_0 = m'_1)$ **の部分** :

入力 : $M_0, G_0, M'_{\bar{b}}, G'_{\bar{b}}, b, r_0, y, g$

出力 : $T_0, W_0, T_1, W_1, e_0, e_1, z_0, z_1$

べき乗演算数 : 6 回

処理内容 :

1. 乱数 R_0, R_1, e を生成する。
2. 以下を計算する。

$$\begin{aligned}(T_b, W_b) &= (y^{R_b}, g^{R_b}) \\ (T_{\bar{b}}, W_{\bar{b}}) &= (y^{R_{\bar{b}}}(M'_{\bar{b}}/M_0)^e, \\ &\quad g^{R_{\bar{b}}}(G'_{\bar{b}}/G_0)^e) \\ e_b &= -e + h(T_0, T_1, W_0, W_1) \\ e_{\bar{b}} &= e \\ z_b &= R_b - e_b r_0 \\ z_{\bar{b}} &= R_{\bar{b}}\end{aligned}$$

$(m_0 m_1 = m'_0 m'_1)$ **の部分** :

入力 : r_0, r_1, y, g

出力 : T, W, z

べき乗演算数 : 2 回

処理内容 :

1. 乱数 R を生成する。
2. 以下を計算する。

$$\begin{aligned}(T, W) &= (y^R, g^R) \\ c &= h(T, W) \\ z &= R - c(r_0 + r_1)\end{aligned}$$

3.2 2^k 入力 MIX-net

前節の 2 入力 2 出力のセルを組み合わせることで、$N(= 2^k, k$: 整数) の入力に対して $NlogN - N + 1$ 個のセルで MIX サーバを構築することができる。

図 3 に 8 入力の場合のセル構成を示す。

3.3 正当性検証

MIX サーバの正当性検証は、セル毎に行い、全てのセルで以下が成立すれば正当である。なお、$c = h(T, W)$ とする。

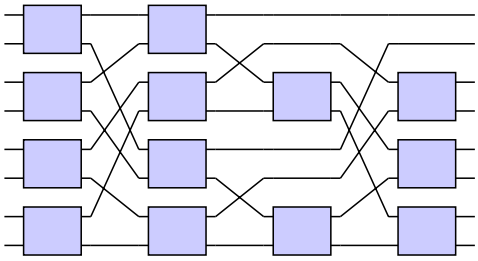

図 3: 8 入力 MIX サーバのセル構成

$$\begin{aligned}e_0 + e_1 &= h(T_0, T_1, W_0, W_1) &\wedge \\ 1 &= y^{z_0}(M'_0/M_0)^{e_0} \times T_0^{-1} &\wedge \\ 1 &= g^{z_0}(G'_0/G_0)^{e_0} \times W_0^{-1} &\wedge \\ 1 &= y^{z_1}(M'_1/M_0)^{e_1} \times T_1^{-1} &\wedge \\ 1 &= g^{z_1}(G'_1/G_0)^{e_1} \times W_1^{-1} &\wedge \\ 1 &= y^{z}(M'_0 M'_1/M_0 M_1)^{c} \times T^{-1} &\wedge \\ 1 &= g^{z}(G'_0 G'_1/G_0 G_1)^{c} \times W^{-1}\end{aligned}$$

4 提案方式

4.1 基本的なアイディア

事前処理化

従来方式では、入力暗号文に依存しないべき乗演算も開票開始後に行なっていた。本提案方式では、これらの処理を開票前に行なうことで開票開始後の演算量を減らす。

ランダム化置換処理と証明生成処理の分割

2.2 節において、現実のシステムでは MIX サーバの不正により、MIX-net から不正な結果が出力される可能性が低いことを述べた。この現実的な前提により、MIX サーバ $j(2 \leq j \leq M)$ は前段の MIX サーバ $j-1$ の出力の正当性検証を行なわず、直ちにランダム化置換処理を開始することができる。また、復号者は MIX サーバ M の出力の正当性検証を行なわずに、復号処理を開始できる。

すなわち、各 MIX サーバは、

1. MIX サーバ j はランダム化置換処理の結果を、MIX サーバ $j+1$ へ渡す。
2. MIX サーバ $j+1$ はランダム化置換処理を開始する。その間、MIX サーバ j は証明生成処理を行なう。

のように連携して処理を行なうことで、最終段の MIX サーバ M がランダム化置換を完了するまで (復号開始まで) の時間を短縮することが可能となる (図 4 参照)。

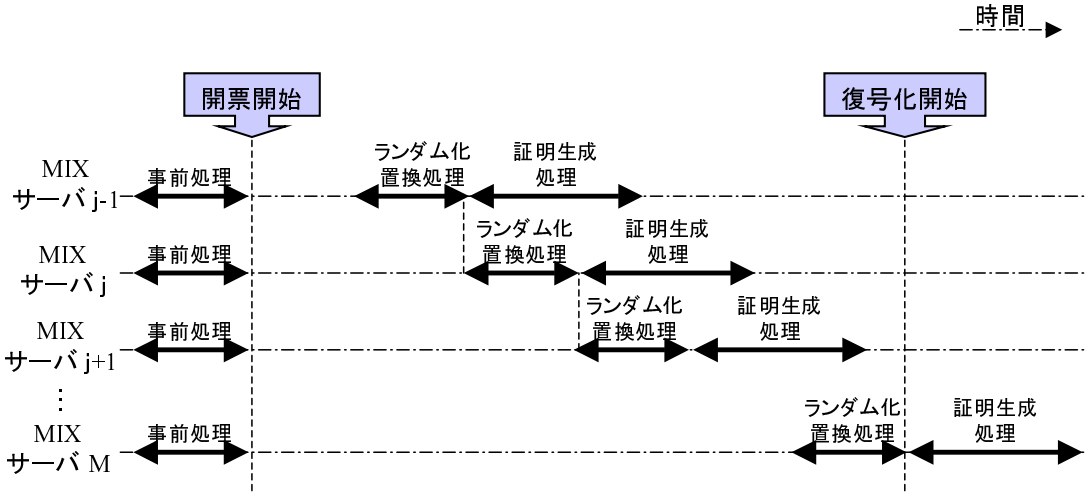

図 4: MIX サーバ連携

4.2 事前処理

各 MIX サーバは事前処理として、入力暗号文に依存しない べき乗演算を開票開始前に行ない、出力を秘密に保管する。なお、$1 \leq i \leq N$, $N = 2^k$, k:整数 とする。

入力 : y, g

出力 : Y_i, G_i
セル毎に、$b, r_0, r_1, R_0, R_1, R, y^{r_0}, y^{r_1}, g^{r_0}, g^{r_1}, y^{R_0}, y^{R_1}, g^{R_0}, g^{R_1}, y^R, g^R$

べき乗演算数 : 12 回

処理内容 :

1. ランダム置換 π を決定する (i 番目の入力暗号文は $\pi(i)$ 番目に出力される)。
   次に π に基づき、各セルの置換パラメータ b を生成する。詳細は文献 [4] 参照。
2. 各セル毎に、以下の秘密乱数を生成する。
$$r_0, r_1, R_0, R_1, R$$
3. 各セル毎に、上記乱数を用いて、以下の値を計算する。
$$y^{r_0}, y^{r_1}, g^{r_0}, g^{r_1}, y^{R_0}, y^{R_1}, g^{R_0}, g^{R_1}, y^R, g^R$$
4. この時点で、置換パラメータ b が決っているため、入力 i が通過するセルを特定できる。さらに、各セルにおいてどのランダム化パラメータ (r_0 もしくは r_1) が適用されるかが分る。そこで、各入力が通過するセルで適用されるランダム化パラメータの和 T_i を求め、$Y_{\pi(i)}, G_{\pi(i)}$ を計算する。
$$Y_{\pi(i)} = y^{T_i}$$
$$G_{\pi(i)} = g^{T_i}$$

4.3 暗号処理

投票者は投票内容 m を、生成元 g、公開鍵 y を用いて以下のように暗号化する。

入力 : m, y, g

出力 : (M, G)

べき乗演算数 : 2 回

処理内容 :

1. 乱数 t を生成する。
2. 以下を計算する。
$$(M, G) = (my^t, g^t)$$

4.4 ランダム化置換処理

各 MIX サーバは、事前処理で算出した値 Y_i, G_i を用いて、入力暗号文 (M_i, G_i) に対して、以下のランダム化置換処理を行なう。なお、以降事前処理で算出した値を **下線** で示す。また、$1 \leq i \leq N$, $N = 2^k$, k: 整数 とする。

入力 : $M_i, G_i, \underline{\mathbf{Y_i}}, \underline{\mathbf{G_i}}$

出力 : M'_i, G'_i

べき乗演算数 : 0 回 (乗算のみ)

処理内容 :

$$M'_{\pi(i)} = M_i \times \underline{\mathbf{Y_i}}$$
$$G'_{\pi(i)} = G_i \times \underline{\mathbf{G_i}}$$

4.5 証明生成処理

各 MIX サーバは、事前処理で算出した値を用いて、証明生成処理を行なう。なお、$1 \leq i \leq N$, $N = 2^k$, k: 整数 とする。

4.5.1 $(m_0 = m'_0 \vee m_0 = m'_1)$ の部分

入力 : セル毎の $M_0, G_0, M'_{\bar{b}}, G'_{\bar{b}}, \underline{\mathbf{b}}, \underline{\mathbf{r_0}}, \underline{\mathbf{R_0}}, \underline{\mathbf{R_1}}, \underline{\mathbf{y}^{\mathbf{R_0}}}, \underline{\mathbf{y}^{\mathbf{R_1}}}, \underline{\mathbf{g}^{\mathbf{R_0}}}, \underline{\mathbf{g}^{\mathbf{R_1}}}$

出力 : $T_0, T_1, W_0, W_1, e_0, e_1, z_0, z_1$

べき乗演算数 : 2 回

処理内容 :

1. セル毎に、乱数 e を生成する。
2. セル毎に、以下を計算する。

$$
\begin{aligned}
(T_b, W_b) &= (\underline{\mathbf{y}^{\mathbf{R_b}}}, \underline{\mathbf{g}^{\mathbf{R_b}}}) \\
(T_{\bar{b}}, W_{\bar{b}}) &= (\underline{\mathbf{y}^{\mathbf{R_{\bar{b}}}}}(M'_{\bar{b}}/M_0)^e, \\
&\quad \underline{\mathbf{g}^{\mathbf{R_{\bar{b}}}}}(G'_{\bar{b}}/G_0)^e) \\
e_b &= -e + h(T_0, T_1, W_0, W_1) \\
e_{\bar{b}} &= e \\
z_b &= \underline{\mathbf{R_b}} - e_b \underline{\mathbf{r_0}} \\
z_{\bar{b}} &= \underline{\mathbf{R_{\bar{b}}}}
\end{aligned}
$$

4.5.2 $(m_0 m_1 = m'_0 m'_1)$ の部分

入力 : セル毎の $\underline{\mathbf{R}}, \underline{\mathbf{y}^{\mathbf{R}}}, \underline{\mathbf{g}^{\mathbf{R}}}, \underline{\mathbf{r_0}}, \underline{\mathbf{r_1}}$

出力 : T, W, z

べき乗演算数 : 0 回

処理内容 :

1. セル毎に、以下を計算する。

$$
\begin{aligned}
(T, W) &= (\underline{\mathbf{y}^{\mathbf{R}}}, \underline{\mathbf{g}^{\mathbf{R}}}) \\
c &= h(T, W) \\
z &= \underline{\mathbf{R}} - c(\underline{\mathbf{r_0}} + \underline{\mathbf{r_1}})
\end{aligned}
$$

5 考察

5.1 安全性

5.1.1 全体検証可能性

任意の第三者は、ゼロ知識証明により MIX-net への入力暗号文の列とその出力暗号文の列が一対一に対応することを確認することができる。

5.1.2 匿名性

全ての MIX サーバが結託する、もしくは 1 人を除く全ての投票者が結託しない限り、入力暗号文と出力暗号文の対応を知ることができない。

表 1: セル毎のべき乗演算数

処理内容	従来方式 (回)	提案方式 (回)
事前処理	-	10
ランダム化 置換処理	12	0
証明生成処理		2

表 2: MIX サーバ毎の処理時間

処理内容	提案方式
事前処理	2 分 43 秒
ランダム化置換	1 秒
証明生成処理	2 分 07 秒

5.2 性能

表 1 に、本稿で提案した各処理のべき乗演算数を示す。また、図 4 に示すように MIX サーバが連携して処理を行なうと、開票開始から復号開始までの処理では、べき乗演算を 0 回にし、乗算のみにすることができる。入力を $N(= 2^k, k$ は整数) とすると、開票開始から全 MIX サーバが証明生成処理を完了させるまで $2(NlogN - N + 1)$ 回のべき乗演算の時間で処理できる。

5.2.1 実装

本節では、本提案方式に基づく電子投票システムの実装評価について述べる。本評価は、Pentium III 1GHz 上で、OEF を用いた楕円暗号実装技術 [3] による 160 bit 楕円 ElGamal 暗号を用い、$4096(=2^{12})$ 票を入力とする。

下記に示す各 MIX サーバにおける提案方式の処理時間を表 2 に示す。

- 事前処理: セル毎に事前処理を行ない、事前処理結果をファイルに出力する
- ランダム化置換処理: ファイルから事前処理結果を入力し、セル毎にランダム化置換を行なう
- 証明生成処理: ファイルから事前処理結果を入力し、セル毎に証明生成処理を行ない、処理後ファイルを削除する

また、上記処理時間及び DB アクセス時間等を含む総処理時間を表 3 に示す。通信時間を無視すると、MIX サーバが 3 台の時、開票開始から以下の時間で各処理が完了する。

- 全 MIX サーバのランダム化置換処理完了: 39 秒 (=(イ) × 3)

表 3: MIX サーバ毎の総処理時間

処理内容	DB アクセス等を含めた総処理時間
(ア) 事前処理	2 分 43 秒
(イ) ランダム化置換	13 秒
(ウ) 証明生成処理	6 分 44 秒
(エ) その他 (復号処理等)	30 秒
正当性検証	10 分 00 秒

- 復号完了: 1 分 9 秒
  (=(イ) × 3 + (エ))
- 全 MIX サーバの証明生成処理完了: 3 分 23 秒
  (=(イ) × 3 + (ウ))

6 まとめ

阿部らの方式をベースに、入力暗号文に依存しない べき乗演算を開票前に処理することで、開票開始後のべき乗演算数を削減可能 (ランダム化置換における べき乗演算は 0 回で、乗算のみで処理可能) にした。さらに、ある MIX サーバが証明生成を処理している間に、次段 MIX サーバがランダム化置換を並列して処理することで、全 MIX サーバの証明処理が完了するまで MIX サーバの台数によらず、開票開始から $2(NlogN - N + 1)$ 回のべき乗演算の時間で処理できる方式を提案した。

また、OEF を用いた楕円暗号実装技術により、MIX サーバが 3 台のとき $4096(= 2^{12})$ 票に対して、開票開始から全 MIX サーバの証明生成処理が完了するまで 3 分 23 秒となることを示した。

参考文献

[1] Masayuki Abe, "Mix-Networks on Permutation Networks," ASIACRYPT'99, LNCS 1716, p.258-273, November, 1999.

[2] 星野文学, 阿部正幸, "Permutation Network を利用した効率的な Mix-net," SCIS2000-B28, 2000 年 1 月.

[3] Kazumaro Aoki, Fumitaka Hoshino, Tetsutaro Kobayashi, "A Cyclic Window Algorithm for ECC Defined over Extension Fields," ICICS 2001, LNCS 2229, p.62-73, November, 2001.

[4] Masayuki Abe, Fumitaka Hoshino, "Remarks on Mix-Network Based on Permutation Networks," PKC2001, LNCS 1992, p.317-324, February, 2001.