Computer Security Symposium 2019
21 - 24 October 2019

Trapdoor DDH 群のある変種の応用について

星野 文学[1,a)]　小林 鉄太郎[1,b)]

概要：対称ペアリングを用いて非対称ペアリングに似た性質を持つ暗号プリミティブを構成できる事が知られている．そのように構成した暗号プリミティブは対称ペアリングと非対称ペアリングを折衷したような性質を持ち，高機能な暗号方式への適用が期待できる．本稿ではそのような暗号プリミティブの応用について考察する．

キーワード：Diffie-Hellman 判定問題, trapdoor DDH 群, 双準同型, 非可換環

On an Application of a Variant of Trapdoor DDH Group

Fumitaka Hoshino[1,a)]　Tetsutaro Kobayashi[1,b)]

Abstract: It is known that a cryptographic primitive like the asymmetric pairing can be constructed using the symmetric one. Since such a primitive has an eclectic nature of the symmetric and asymmetric pairings, it is expected that the primitive can be available to construct highly functional schemes. In this paper, we consider some applications of such a cryptographic primitive.

Keywords: decisional Diffie-Hellman problem, trapdoor DDH group, bihomomorphism, non-commutative ring

1. はじめに

素数位数巡回群 $\mathbb{G}$ を対称ペアリング群とし，$e:\mathbb{G}\times\mathbb{G}\to\mathbb{G}_T$ をペアリングとする．$\mathbb{G}$ は $(\mathbb{Z}/q\mathbb{Z})^+$ と同型であるから，これを $\mathbb{F}_q$ だと思えば，群演算を和，冪をスカラー倍として，$\mathbb{G}$, は階数 1 の $\mathbb{F}_q$ ベクトル空間と見做すことができる．従って $\mathbb{G}':=\mathbb{G}\oplus\mathbb{G}$ は成分毎の群演算を和，成分毎の冪をスカラー倍とした階数 2 の $\mathbb{F}_q$ ベクトル空間と見做すことができる．g を $\mathbb{G}$ の生成元として $a,b\in\mathbb{F}_q^*$ をランダムに選ぶと $g_1:=(g^a,g^b)\in\mathbb{G}'$ は位数 q の巡回群 $\mathbb{G}_1=\langle g_1\rangle$ を生成する．同様に $c,d\in\mathbb{F}_q^*$ をランダムに選ぶと $g_2:=(g^c,g^d)\in\mathbb{G}'$ は高い確率で位数 q の巡回群 $\mathbb{G}_2=\langle g_2\rangle\neq\mathbb{G}_1$ を生成する．さらに $\gamma,\gamma'\in\mathbb{F}_q$ を適当な定数として $e':\mathbb{G}_1\times\mathbb{G}_2\to\mathbb{G}_T$ を

$$e':(f_1,f_2),(h_1,h_2)\mapsto e(f_1,h_2)^{\gamma}e(h_1,f_2)^{\gamma'}$$

と定義すれば，e' は非退化双線形写像の確率的多項式時間アルゴリズムとなる．従って $(\mathbb{G}_1,\mathbb{G}_2)$ は安全かどうかは別として，非対称ペアリング群と見なすことが出来る．

今，上記の a,b,c,d $(\Delta:=ad-bc\neq 0)$ が予め分かっている場合，ベクトル空間 $\mathbb{G}':=\mathbb{G}\oplus\mathbb{G}$ の任意の元 $v:=(v_1,v_2)$ は $\mathbb{G}_1$ の元と $\mathbb{G}_2$ の元

$$\begin{pmatrix}({v_1}^d\cdot{v_2}^{-c})^{\Delta^{-1}\cdot a},({v_1}^d\cdot{v_2}^{-c})^{\Delta^{-1}\cdot b}\end{pmatrix}\in\mathbb{G}_1,\\ \begin{pmatrix}(v_1^{-b}\cdot{v_2}^a)^{\Delta^{-1}\cdot c},(v_1^{-b}\cdot{v_2}^a)^{\Delta^{-1}\cdot d}\end{pmatrix}\in\mathbb{G}_2 \qquad (1)$$

の和 (成分毎の群演算) に多項式時間で分解できる．では a,b,c,d が明かされない場合，このような分解は簡単であろうか? 吉田らは上記のベクトル分解問題を $\mathbb{G}$ 上の CDH 問題へ帰着し，a,b,c,d を落し戸として使用する暗号プロトコルへの応用を示した [16]．この方法論はその後発展し，様々な解析や応用が検討されている [1–3, 5, 10–13]．

ところで，80 年代に静谷らは，概ね次のような離散対数

[1] NTT セキュアプラットフォーム研究所
NTT Secure Platform Laboratories
[a)] fumitaka.hoshino.bt@hco.ntt.co.jp
[b)] tetsutaro.kobayashi.dr@hco.ntt.co.jp

の自然な拡張を考案した [14].

- $\langle g\rangle$ を素数位数巡回群とし g をその生成元とし, その位数 (素数) を q とする. 写像 : $\langle g\rangle^{n\times m} \to \mathbb{F}_q^{n\times m}$,

$$\begin{pmatrix} g^{x_{11}} & \dots & g^{x_{1m}} \\ \vdots & \ddots & \vdots \\ g^{x_{n1}} & \dots & g^{x_{nm}} \end{pmatrix} \mapsto \begin{pmatrix} x_{11} & \dots & x_{1m} \\ \vdots & \ddots & \vdots \\ x_{n1} & \dots & x_{nm} \end{pmatrix}$$

を考える. $\mathbb{F}_q$ 行列 $x := (x_{ij}) \in \mathbb{F}_q^{n\times m}$ を g を底とする $X := (g^{x_{ij}}) \in \langle g\rangle^{n\times m}$ の離散対数と呼び, X を g^x と書く.

- $x, y \in \mathbb{F}_q^{n\times m}$, $X := g^x$, $Y := g^y$ とする. 積 XY を

$$\begin{array}{rccccc} XY: & \langle g\rangle^{n\times m} & \times & \langle g\rangle^{n\times m} & \to & \langle g\rangle^{n\times m}, \\ & g^x & , & g^y & \mapsto & g^{x+y}, \end{array}$$

と定義する. 積 XY は可換.

- $x \in \mathbb{F}_q^{n\times \ell}$, $y \in \mathbb{F}_q^{\ell\times m}$ とし, $X := g^x$, $Y := g^y$ とする. 双準同型 X^y を

$$\begin{array}{rccccc} X^y: & \langle g\rangle^{n\times \ell} & \times & \mathbb{F}_q^{\ell\times m} & \to & \langle g\rangle^{n\times m}, \\ & g^x & , & y & \mapsto & g^{xy}, \end{array}$$

双準同型 ${}^xY$ を

$$\begin{array}{rccccc} {}^xY: & \mathbb{F}_q^{n\times \ell} & \times & \langle g\rangle^{\ell\times m} & \to & \langle g\rangle^{n\times m}, \\ & x & , & g^y & \mapsto & g^{xy}, \end{array}$$

と定義する. X^y を右冪乗 ${}^xY$ を左冪乗と呼ぶ.

- X や Y の離散対数を知らなくとも $\langle g\rangle$ 上の群演算を用いて右冪乗, 左冪乗および積は効率的に計算可能.

このような概念を用いると, 例えば式 (1) のベクトル分解は

$$(\phi_1(v^{A^{-1}}))^A \in \mathbb{G}_1,$$
$$(\phi_2(v^{A^{-1}}))^A \in \mathbb{G}_2$$

のように見通し良く記述できる. ここで $\phi_1 : (v_1, v_2) \mapsto (v_1, 1)$, $\phi_2 : (v_1, v_2) \mapsto (1, v_2)$ は射影演算で $A := \begin{pmatrix} a & b \\ c & d \end{pmatrix}$ とする. g^A が公開されている状況で, 離散対数 A を知っていればこの分解は簡単だが, 知らないものがこの分解を行う事は容易ではなさそうなので, A を落とし戸とすることが可能である. このような概念を幾らか拡張すると, 非対称ペアリング群と対称ペアリング群を折中したような特殊な暗号プリミティブが得られることが知られている [7]. 本論文では, この暗号プリミティブの応用について考察する.

2. 準備

2.1 離散対数

計算量的暗号方式の設計においては, 離散対数とは様々な暗号学的応用が可能な暗号プリミティブ (原始方式) の事であり, 形式的には安全変数 $\lambda \in \mathbb{N}$ を入力とし, λ でパラメタライズされる安全性を満たす (と思しき), ある代数的構造の効率的な符号に関する記述 $(\mathbb{L}, \mathbb{G}, \mathsf{aux})$ を出力する確率的多項式時間アルゴリズム

$$\mathcal{G} : 1^\lambda \xmapsto{\$} (\mathbb{L}, \mathbb{G}, \mathsf{aux})$$

であると定義される. $\mathbb{L}, \mathbb{G}$ は代数的構造の効率的な符号化方法を記述する文字列であるが, くどいので以降は $\mathbb{L}, \mathbb{G}$ と代数的構造とを同一視する. 従って $|\mathbb{L}|$ や $|\mathbb{G}|$ 等と記述した時, それは記述の長さや符号語の数等ではなくて, 記述された代数的構造の位数を意味するとする. 典型的には $\mathbb{G}$ を素数位数巡回群 $\langle g\rangle$ (位数 q) とし $\mathbb{L} := \mathbb{F}_q$ とされるが, ここではその拡張を扱うので,

1. $\mathbb{L}$ を単位的環, $\mathbb{G}$ をその環上の加群とする. また $\mathsf{aux} \in \{0,1\}^*$ は補助情報とする.

情報理論的な安全性に関する要請により少なくとも

2. $|\mathbb{L}|, |\mathbb{G}| \geq 2^{\Theta(\lambda)}$

が必要である. 典型的な離散対数とのアナロジーにより加群としての和とスカラー倍を $\mathbb{G}$ 上の積および冪乗と呼び, 記法も巡回群の積および冪乗に準じる. 一般の $\mathbb{L}$ について左右 2 種類の冪乗が存在するが, $\mathbb{L}$ が可換の場合には両者は一致する. これから, この拡張に合わせて離散対数の概念を幾分精密に定義していくが, この定義は $\mathbb{G} := \langle g\rangle$, $\mathbb{L} := \mathbb{F}_q$ とすれば典型的な離散対数の定義と一致する.

3. 次の λ に関する確率的多項式時間アルゴリズムが自明であるか, あるいは $(\mathbb{L}, \mathbb{G}, \mathsf{aux})$ の何れかに含まれる.
   - $\mathbb{L}, \mathbb{G}$ 上の標本.
   - $\mathbb{L}, \mathbb{G}$ の元の識別 $(=, \neq)$.
   - $\mathbb{L}$ 上の環演算.
   - $\mathbb{G}$ 上の積 : $\mathbb{G} \times \mathbb{G} \to \mathbb{G}$.
   - $\mathbb{G}$ 上の右冪乗 (双準同型) : $\mathbb{G} \times \mathbb{L} \to \mathbb{G}$.
   - $\mathbb{G}$ 上の左冪乗 (双準同型) : $\mathbb{L} \times \mathbb{G} \to \mathbb{G}$.

一般に暗号認証技術の設計において安全性の証明を行う際には, 暗号プリミティブに対して定義される何らかの暗号学的問題を考察する. λ は安全変数と呼ばれ, 暗号学的問題を解こうと試みる確率的多項式時間攻撃者 $\mathcal{A}$ に対して定義される利得が, 如何なる $\mathcal{A}$ に対しても λ に関して無視可能となる事が暗号プリミティブが安全である事の差し当たっての定義である. 暗号プリミティブの安全性を数学的に証明する事は多くの場合は困難であり, 通常は経験的に成立すると予想される仮説が用いられる. そのような仮説を暗号学的仮定と呼ぶ. 通常, 離散対数プリミティブにおいては暗号学的仮定は少なくとも次の仮定を含意する.

4. $\mathcal{G}$-離散対数仮定: オラクルチューリングマシン

$\mathrm{Exp}^{\mathcal{G},\mathcal{A}}(1^\lambda) :=$

$(\mathbb{L}, \mathbb{G}, \mathsf{aux}) \xleftarrow{\$} \mathcal{G}(1^\lambda)$,
$x \xleftarrow{\$} \mathbb{L},\ g \xleftarrow{\$} \mathbb{G},\ y \leftarrow g^x$,
$x^* \xleftarrow{\$} \mathcal{A}(\mathbb{L}, \mathbb{G}, \mathsf{aux}, g, y)$,
$\mathtt{Output}\ (y \stackrel{?}{=} g^{x^*})$.

に対し定義される利得 $\mathsf{Adv}^{\mathcal{G},\mathcal{A}}(\lambda) := \Pr[\mathrm{Exp}^{\mathcal{G},\mathcal{A}}(1^\lambda) = 1]$ が如何なる確率的多項式時間チューリングマシン $\mathcal{A}$ に対しても無視可能である.

概ね底をランダムに一つ選んだ時に冪を変数とした冪乗関数の原像困難性を言っている. (厳密には左右の冪乗の両方に対して離散対数問題を考える事が出来るが, 本稿で扱う具体例については双方向に帰着がある.) 離散対数仮定が尤もらしい $\mathcal{G}$ を離散対数群と呼ぶ. くどいので $\mathcal{G}$ が離散対数群である事を $\mathbb{G}$ が離散対数群であるとも言う. 膨大な量の離散対数仮定を含意する仮定 (眷属) が提案されており [15], その中でも次の computational Diffie-Hellman (CDH) 仮定が特に有名である.

$\mathcal{G}$-CDH 仮定: オラクルチューリングマシン
$\mathrm{Exp}^{\mathcal{G},\mathcal{A}}(1^\lambda) :=$

$(\mathbb{L}, \mathbb{G}, \mathsf{aux}) \xleftarrow{\$} \mathcal{G}(1^\lambda)$,
$g_0 \xleftarrow{\$} \mathbb{G}$,
$g_1 \leftarrow {}^a g_0 \mid a \xleftarrow{\$} \mathbb{L}$,
$g_2 \leftarrow g_0^b \mid b \xleftarrow{\$} \mathbb{L}$,
$g_3 \xleftarrow{\$} \mathcal{A}(\mathbb{L}, \mathbb{G}, \mathsf{aux}, g_0, g_1, g_2)$,
$\mathtt{Output}\ (g_3 \stackrel{?}{=} {}^a g_0^b)$.

に対し定義される利得 $\mathsf{Adv}^{\mathcal{G},\mathcal{A}}(\lambda) := \Pr[\mathrm{Exp}^{\mathcal{G},\mathcal{A}}(1^\lambda) = 1]$ が如何なる確率的多項式時間チューリングマシン $\mathcal{A}$ に対しても無視可能である.

CDH 仮定が尤もらしい $\mathcal{G}$ あるいは $\mathbb{G}$ を CDH 群と呼ぶ. $\mathbb{G}$ に有限体の乗法群を用いたものや有限体上定義された楕円曲線を用いたものが CDH 群の有名な例である. 一方, 同じ代数的構造 (巡回群) でも $\mathbb{L} = \mathbb{F}_q$, $\mathbb{G} = (\mathbb{Z}/q\mathbb{Z})^+$ とすれば, その CDH 問題 (および離散対数問題) は自明に解くことが出来る. こうした仮定が困難そうであるか, あるいは明らかに簡単であるかは $\mathbb{L}$ や $\mathbb{G}$ をどのように符号化するかに依存する.

2.2 ペアリング

ペアリングとは概ね次のような離散対数の拡張 (確率的多項式時間アルゴリズム) である.

$$\mathcal{G}' : 1^\lambda \xmapsto{\$} (\mathbb{L}, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, \mathsf{aux}')$$

1. $\forall x \in \{1, 2, T\}$ に関してオラクルチューリングマシン $\mathcal{G}_x^{\mathcal{G}'}(1^\lambda) :=$

   $(\mathbb{L}, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, \mathsf{aux}') \xleftarrow{\$} \mathcal{G}'(1^\lambda)$,
   $\mathsf{aux} \leftarrow (\mathbb{L}, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, \mathsf{aux}')$,
   $\mathtt{Output}(\mathbb{L}, \mathbb{G}_x, \mathsf{aux})$.

   が全て CDH 群である. 即ち $\mathbb{G}_x$ が全て CDH 群である.
2. 次の λ に関する確率的多項式時間アルゴリズムが自明であるか, あるいは $(\mathbb{L}, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, \mathsf{aux}')$ の何れかに含まれる.
   - ペアリング (双準同型) $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$.

$\mathbb{G}_1, \mathbb{G}_2$ を入力群 (source group), $\mathbb{G}_T$ を出力群 (target group) と呼ぶ. Galbraith らは, 暗号方式に用いられるペアリングを大雑把に以下の 3 つの型に分類した [4].

Type 1: $\mathbb{G}_1 = \mathbb{G}_2$

Type 2: $\mathbb{G}_1 \neq \mathbb{G}_2$, $\phi : \mathbb{G}_2 \to \mathbb{G}_1$ なる多項式時間同型写像が存在する.

Type 3: $\mathbb{G}_1 \neq \mathbb{G}_2$, $\mathbb{G}_1, \mathbb{G}_2$ の間に多項式時間同型写像が存在しない.

一般に Type 1 を対称ペアリングと呼び, Type 2, Type 3 を非対称ペアリングと呼ぶ. CDH 群 $\mathcal{G}$ に対して次の仮定は decisional Diffie-Hellman (CDH) 仮定と呼ばれる.

$\mathcal{G}$-DDH 仮定: オラクルチューリングマシン
$\mathrm{Exp}^{\mathcal{G},\mathcal{A}}(1^\lambda) :=$

$(\mathbb{L}, \mathbb{G}, \mathsf{aux}) \xleftarrow{\$} \mathcal{G}(1^\lambda)$,
$g_0 \xleftarrow{\$} \mathbb{G}$,
$g_1 \leftarrow {}^a g_0 \mid a \xleftarrow{\$} \mathbb{L}$,
$g_2 \leftarrow g_0^b \mid b \xleftarrow{\$} \mathbb{L}$,
$h_0 \leftarrow g_0^c \mid c \xleftarrow{\$} \mathbb{L}$,
$h_1 \leftarrow {}^a g_0^b$,
$g_3 \leftarrow h_d \mid d \xleftarrow{\$} \{0, 1\}$,
$d^* \xleftarrow{\$} \mathcal{A}(\mathbb{L}, \mathbb{G}, \mathsf{aux}, g_0, g_1, g_2, g_3)$,
$\mathtt{Output}\ (d \stackrel{?}{=} d^*)$.

に対し定義される利得 $\mathsf{Adv}^{\mathcal{G},\mathcal{A}}(\lambda) := |\Pr[\mathrm{Exp}^{\mathcal{G},\mathcal{A}}(1^\lambda) = 1] - 1/2|$ が如何なる確率的多項式時間チューリングマシン $\mathcal{A}$ に対しても無視可能である.

DDH 仮定が尤もらしい $\mathcal{G}$ あるいは $\mathbb{G}$ を DDH 群と呼ぶ. 多項式時間双準同型 e の存在により, $\mathbb{L}$ が可換の時はペアリングの入力群に関して次の事が自明に分かる.

Type 1 では $\mathbb{G}_1 = \mathbb{G}_2$ 上で DDH 仮定は成立しない.

Type 2 では $\mathbb{G}_2$ 上で DDH 仮定は成立しない.

それ以外の $\mathbb{G}_x$ では DDH 仮定が成立していてもペアリングの形式的な定義とは矛盾しないので, そのような仮定, 例えば SXDH 仮定などはプロトコルの設計にしばしば用いられる. また $\mathbb{L}$ が非可換であるときは DDH 仮定とペアリングの形式的な定義とは矛盾しない. $\mathbb{L}$ が非可換であるときの $\mathbb{G}_x$ の DDH 仮定の応用が本稿のテーマである.

2.3 Trapdoor DDH

Trapdoor DDH 群とは落とし戸があれば DDH 仮定を破ることができる DDH 群の拡張で, 本稿で扱うのは, 次の 2 つの確率的多項式時間アルゴリズムが存在するものである.

- $\mathbb{G}$ の元および対応する落とし戸の組をランダムに出力する確率的多項式時間アルゴリズム

$$\begin{array}{rccccc} \texttt{tsamp}: & 1^* & \xrightarrow{\$} & \mathbb{G} & \times & \{0,1\}^* \\ & 1^\lambda & \mapsto & g_0 & , & t \end{array}$$

- 上記 tsamp によって生成された g_0 を用いて生成された DDH インスタンス $(\mathbb{L}, \mathbb{G}, \mathsf{aux}, g_0, {}^a g_0, g_0^b, g_3)$ と対応する落とし戸 t を入力として, $(g_3 \stackrel{?}{=} {}^a g_0^b)$ であるか否かを出力する確率的多項式時間アルゴリズム

$$\texttt{solve} : (\mathbb{L}, \mathbb{G}, \mathsf{aux}, g_0, {}^a g_0, g_0^b, g_3), t \xrightarrow{\$} (g_3 \stackrel{?}{=} {}^a g_0^b)$$

2.4 Trapdoor DDH の具体的な構成

[7] にはペアリングを拡張した特殊な trapdoor DDH 群の具体的構成が示されている. この trapdoor DDH 群の具体的構成を要約すると, およそ次のようになる.

- $\langle g \rangle$ を (通常の) 対称ペアリングの入力群とし, g をその生成元とする. 同様に $\langle g_T \rangle$ を対応する出力群とし, g_T をその生成元とする.
  - $\mathbb{G} := \langle g \rangle^{n \times n}$.
  - $\mathbb{L} := \mathbb{F}_q^{n \times n}$.
  - $\mathbb{G}_T := \langle g_T \rangle^{n \times n}$.

$\mathbb{G}$ は成分毎の群演算を群演算とするアーベル群で, これを trapdoor DDH 群と見なす. 静谷らの定義と同様に非可換環 $\mathbb{L}$ を $\mathbb{G}$ に対する離散対数と見なし, 積や冪乗も同様に定義する. また, 双準同型 e を

$$\begin{array}{rccccc} e: & \mathbb{G} & \times & \mathbb{G} & \to & \mathbb{G}_T, \\ & g^x & , & g^y & \mapsto & g_T^{xy}, \end{array}$$

と定義する. 離散対数 x や y を知らなくとも $\langle g \rangle$ 上のペアリングを用いて e は効率的に計算可能である. e を改めてペアリングと呼ぶ. このようなペアリング関数 e の拡張は暗号プロトコルの設計の分野で良く知られており, 既に多用されている [6]. $\mathbb{G}$ 上の冪乗やペアリングを使って, tsamp および solve を次のように構成する.

$\texttt{tsamp}(1^\lambda) :=$
$t \xleftarrow{\$} \mathbb{L}^*$,
$g_0 \leftarrow g^t$,
Output (g_0, t).

$\texttt{solve}((g_0, g_1, g_2, g_3), t) :=$
Output $(e(g_1^{t^{-1}}, g_2) \stackrel{?}{=} e(I, g_3))$.

但し $\mathbb{L}^*$ は $\mathbb{L}$ の正則元 (非零因子) の集合とし i を $\mathbb{L}$ 上の単位行列として $I = g^i$ とする. また $\mathbb{L}^*$ 上の乗法逆元を計算する確率的多項式時間アルゴリズムが必要であるが, $\mathbb{L}^* = \mathrm{GL}(n, \mathbb{F}_q)$ 上の乗法逆元は効率的に計算可能なので問題ない. 厳密には $\mathbb{G}$ の分布と $\mathbb{G}^* = \mathrm{GL}(n, \langle g \rangle) := g^{\mathbb{L}^*}$ の分布は異なるが, $g_0 \in \langle g \rangle^{n \times n}$ が正則であるか否か判定する問題を考えれば, それが $\langle g \rangle$ 上の DLIN 仮定の亜種であるという事が直ちに分かる. また t を知らない攻撃者に対する, このプリミティブの DDH 安全性は行列 $\begin{pmatrix} g_0 & g_1 \\ g_2 & g_3 \end{pmatrix} \in \langle g \rangle^{2n \times 2n}$ (の離散対数) が正則であるか否かという問題を考えれば, やはり $\langle g \rangle$ 上の DLIN 仮定の亜種であるという事が直ちに分かる.

3. 暗号プロトコルへの応用

前述の定義によれば $\mathbb{L}$ が非可換でも, $\mathbb{L}$ の非可換性や冪乗に左右がある事に目を瞑れば, 離散対数やペアリングの定義はそれほど大きな変更を迫られない上に, $\mathbb{G}$ 上の DDH 仮定に対して落とし戸を構成できるという新しい機能を追加できることが分かった. 従って, 既存の暗号プロトコルで使用されてきた典型的な離散対数群やペアリング群を前述の構成で置き換える事が出来れば, プロトコルにに新たな機能を付加できるかもしれない. 本節ではその可能性を検討するため, $\mathbb{L}$ の非可換性が主要な暗号プロトコルにどのような影響を与えるかを調べる.

3.1 鍵交換

Diffie-Hellman 鍵交換は幾分対称性が失われるが, $\mathbb{L}$ が非可換でも問題なく実行できる.

- Alice が $a \in \mathbb{L}$ を生成し, ${}^a g$ を Bob に送信.
- Bob が $b \in \mathbb{L}$ を生成し, g^b を Alice に送信.
- Alice は $K_a = {}^a(g^b)$ を計算.
- Bob は $K_b = ({}^a g)^b$ を計算.
- K_a と K_b は同じ値となる.

Alice	$(\mathbb{L}, \mathbb{G}, \mathsf{aux}) \xleftarrow{\$} \mathcal{G}(\lambda),\ g \xleftarrow{\$} \mathbb{G}$	**Bob**
$a \xleftarrow{\$} \mathbb{L}$		$b \xleftarrow{\$} \mathbb{L}$
$A \leftarrow {}^a g$	$\xrightarrow{\quad A \quad}$	$B \leftarrow g^b$
$K_a \leftarrow {}^a B$	$\xleftarrow{\quad B \quad}$	$K_b \leftarrow A^b$
	$(K_a = {}^a g^b = K_b)$	

図 1 DH 鍵交換

3.2 知識証明

Schnorr 認証は $\mathbb{L}$ が非可換でもあまり問題なく実行できる.

- Prover は秘密鍵 $x \in \mathbb{L}$ を生成し, 公開鍵 $y = g^x$ を公開する.
- Prover は $t \in \mathbb{L}$ を生成し, コミット $T = g^t$ を Verifier に送信.
- Verifier はチャレンジ $c \in \mathbb{L}$ を生成し Prover に送信.
- Prover はレスポンス $s = t - xc$ を Verifier に送信.
- Verifier は $T \stackrel{?}{=} g^s y^c$ を調べる.

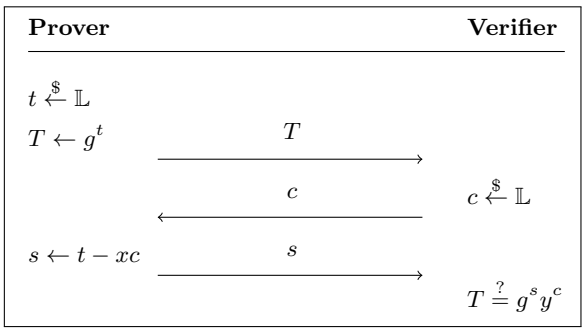

図 2 $PK\{(x) : y = g^x\}$

ゼロ知識性に関しては何の問題も無い. special soundness を証明する為には, ふたつのチャレンジ $c_1, c_2 \in \mathbb{L}$ を $c_1 - c_2 \in \mathbb{L}^*$ となるよう選ぶ必要がある.

4. 匿名化可能署名への応用

前節の解析によれば比較的単純な暗号プロトコルであれば $\mathbb{L}$ が非可換でも問題なく実装できそうであった. ここでは, もう少し大きなプロトコルにこの方法論を適用して, 既存プロトコルに新たな機能を導入する事を考える.

4.1 匿名化可能署名

匿名化可能署名 (Anonymizable Signature) [8] とはリング署名の拡張であり, 匿名化可能署名を用いると署名されたメッセージを持つ者は, 誰でも後からその署名を匿名署名に変換できる. 即ち, 署名者は適切なエージェントに署名を渡しておけば, 後で自分が匿名化に関与する必要が無い. $k \in \mathbb{N}$ を安全パラメタとし, $N = \{0, 1, \cdots\}$ を署名者集合とする. 署名者 i の秘密鍵を x_i, 公開鍵を y_i とする. 署名者の部分集合 $L \subset N$ はリングと呼ばれる. 記述を簡単にするため, リング L に対して a_L を $a_L = (a_i)_{i \in L}$ と定義する. 匿名化可能署名方式 Σ は次の構文を満たす 4 つの確率的多項式時間アルゴリズム ($\mathtt{KeyGen}, \mathtt{Sign}, \mathtt{Anonymize}, \mathtt{Verify}$) により構成される.

鍵生成アルゴリズム $\mathtt{KeyGen}(1^k) \xrightarrow{\$} (x, y)$: は安全パラメタ 1^k を入力とし秘密鍵 x および公開鍵 y を出力とする確率的多項式時間アルゴリズム.

署名アルゴリズム $\mathtt{Sign}(x, m) \xrightarrow{\$} r$: は秘密鍵 x およびメッセージ m を入力とし署名 r を出力とする確率的多項式時間アルゴリズム.

匿名化アルゴリズム $\mathtt{Anonymize}(i, r, L, y_L, m) \xrightarrow{\$} \sigma/\perp$: は署名者 ID i, 署名 r, リング $L \subset N$, 公開鍵リスト y_L, およびメッセージ m を入力とし, リング署名 σ または拒絶 $\perp$ を出力とする確率的多項式時間アルゴリズム.

検証アルゴリズム $\mathtt{Verify}(L, y_L, m, \sigma) \xrightarrow{\$} 0/1$: はリング $L \subset N$, 公開鍵リスト y_L, メッセージ m, およびリング署名 σ, を入力とし, 単一ビット $b \in \{0, 1\}$ を出力とする確率的多項式時間アルゴリズム.

構文 : 如何なる (多項式長の) メッセージ $m \in \{0,1\}^*$, 如何なる (多項式長の) リング $L \subset N$, および如何なる署名者 $i \in L$ に対しても

$$\Pr\left[\, b = 0 \;\middle|\; \begin{array}{l} \forall j \in L, (x_j, y_j) \xleftarrow{\$} \mathtt{KeyGen}(1^k), \\ r \xleftarrow{\$} \mathtt{Sign}(x_i, m), \\ \sigma \xleftarrow{\$} \mathtt{Anonymize}(r, L, y_L, m), \\ b \xleftarrow{\$} \mathtt{Verify}(L, y_L, m, \sigma) \end{array} \right]$$

が k に関して無視可能.

署名者の匿名性を壊さないよう署名 r は署名者とエージェントの間の秘密としなければならない.

4.2 エージェント指定匿名化可能署名

匿名化可能署名において, 署名者がエージェントに渡す署名は通常の署名であり, エージェントはこの署名を持つ事以外は, 第三者と何ら違いは無い. 従って, 攻撃者が漏洩などによりこの署名を入手した場合, エージェントと同様に署名者がこの署名を行った事を確信できる. 従ってエージェントが署名者から託された署名は秘密として安全に管理する必要がある. 通常エージェントは沢山の署名者から沢山の署名を預かると考えて良く, 集める署名の数に比例して安全に管理すべき秘密の量が大きくなる. エージェント指定匿名化可能署名 (Designated Agent Anonymizable Signature) [9] とは匿名化可能署名の拡張であり, この問題を解決するために考案された. エージェント指定匿名化可能署名 Σ は次の構文を満たす 5 つの確率的多項式時間アルゴリズム ($\mathtt{Setup}, \mathtt{KeyGen}, \mathtt{Sign}, \mathtt{Anonymize}, \mathtt{Verify}$) により構成される.

エージェント鍵生成アルゴリズム $\mathtt{Setup}(1^k) \xrightarrow{\$} (w, \rho)$: は安全パラメタ 1^k を入力としエージェント秘密鍵 w および公開パラメタ ρ を出力とする確率的多項式時間アルゴリズム.

鍵生成アルゴリズム $\mathtt{KeyGen}(\rho) \xrightarrow{\$} (x, y)$: は公開パラメタ ρ を入力とし秘密鍵 x および公開鍵 y を出力とす

る確率的多項式時間アルゴリズム.

署名アルゴリズム $\mathtt{Sign}(x,m) \xrightarrow{\$} r$: は秘密鍵 x およびメッセージ m を入力とし署名 r を出力とする確率的多項式時間アルゴリズム.

匿名化アルゴリズム $\mathtt{Anonymize}(i,r,L,y_L,m) \xrightarrow{\$} \sigma/\perp$: は署名者 ID i, 署名 r, リング $L \subset N$, 公開鍵リスト y_L, およびメッセージ m を入力とし, リング署名 σ または拒絶 $\perp$ を出力とする確率的多項式時間アルゴリズム.

検証アルゴリズム $\mathtt{Verify}(L,y_L,m,\sigma) \xrightarrow{\$} 0/1$: はリング $L \subset N$, 公開鍵リスト y_L, メッセージ m, およびリング署名 σ, を入力とし, 単一ビット $b \in \{0,1\}$ を出力とする確率的多項式時間アルゴリズム.

構文 : 如何なる (多項式長の) メッセージ $m \in \{0,1\}^*$, 如何なる (多項式長の) リング $L \subset N$, および如何なる署名者 $i \in L$ に対しても

$$\Pr\left[\, b=0 \;\middle|\; \begin{array}{l} (w,\rho) \xleftarrow{\$} \mathtt{Setup}(1^k), \\ \forall j \in L, (x_j,y_j) \xleftarrow{\$} \mathtt{KeyGen}(\rho), \\ r \xleftarrow{\$} \mathtt{Sign}(x_i,m), \\ \sigma \xleftarrow{\$} \mathtt{Anonymize}(w,r,L,y_L,m), \\ b \xleftarrow{\$} \mathtt{Verify}(L,y_L,m,\sigma), \end{array} \right]$$

が k に関して無視可能.

署名者の匿名性を壊さないよう署名 r は署名者とエージェントの間の秘密としなければならないが, r が攻撃者に漏洩しても, エージェント秘密鍵 w が漏洩していなければ r が どの x_i によって作成された署名なのかは計算量的に識別困難とする.

4.3 エージェント指定匿名化可能署名の具体的構成

例えば次のような Trapdoor-DDH 群を用いたエージェント匿名化可能署名の具体的構成を考えることが出来る. $\mathbb{G}$ を上記 Trapdoor-DDH 群とし $\mathbb{L}$ をその離散対数とし, $\mathbb{G}_T = \mathrm{GL}(n, \langle e(\alpha,\alpha)\rangle)$ とする. $e : \mathbb{G}\times\mathbb{G} \to \mathbb{G}_T$ を上記 Trapdoor-DDH 群のペアリングとする. $H : \{0,1\}^* \to \mathbb{G}$ および $H' : \{0,1\}^* \to \mathbb{L}^3$ を互いに独立なランダムオラクルとする. $\rho = (\mathbb{L},\mathbb{G},\mathbb{G}_T,e,g,H,H')$ を公開パラメタと呼ぶ.

$\mathtt{Setup}(1^k)$:
$w \xleftarrow{\$} \mathbb{L}^*$
$g \leftarrow I^w$
$\rho \leftarrow (\mathbb{L},\mathbb{G},\mathbb{G}_T,e,g,H,H')$
return (w,ρ)

$\mathtt{KeyGen}(\rho)$:
$(\mathbb{L},\mathbb{G},\mathbb{G}_T,e,g,H,H') \leftarrow \rho$
$x \xleftarrow{\$} \mathbb{L},\ y \leftarrow {}^x g$
return (x,y)

$\mathtt{Sign}(x,m)$:
return $\sigma \leftarrow {}^x H(\rho,m)$

$\mathtt{Anonymize}(i,r,L,y_L,m)$:
$h \leftarrow H(\rho,m)$
if $e(I,r) \neq e(y_i^{w^{-1}},h)$ return $\perp$
$(\exists j \in L, e(I,r) = e(y_j^{w^{-1}},h))$ なる $r, y_j^{w^{-1}}$ の知識証明を次のように生成する:
$t_1,t_2,t_3,\beta \xleftarrow{\$} \mathbb{L}$
$V \leftarrow {}^{\beta} e(I,y_i),\ U \leftarrow {}^{\beta} e(I,r)$
$T_{1,i} \leftarrow {}^{t_1} e(I,y_i)$
$\forall j \in L \setminus \{i\}$
$c_{1,j}, z_{1,j} \xleftarrow{\$} \mathbb{L}$
$T_{1,j} \leftarrow e(I, {}^{z_{1,j}} y_j)\ {}^{c_{1,j}} V$
$T_2 \leftarrow {}^{t_2} e(I,g),\ T_2' \leftarrow {}^{t_2} e(I,h)$
$T_3 \leftarrow {}^{t_3} e(I,I)$

$(c_{1,i},c_2,c_3) \leftarrow$
$H'(\rho,L,m,y_L,U,V,T_{1,L},T_2,T_2',T_3)$
$-\sum_{j\neq i}(c_{1,j},0,0)$
$z_{1,i} \leftarrow t_1 - c_{1,i}\beta$
$z_2 \leftarrow {}^{t_2} I/^{c_2\beta y_i^{w^{-1}}}$
$z_3 \leftarrow {}^{t_3} I/^{c_3\beta r}$

return $\sigma \leftarrow (U,V,c_{1,L},c_2,c_3,z_{1,L},z_2,z_3)$

$\mathtt{Verify}(L,y_L,m,\sigma)$:
$(U,V,c_{1,L},c_2,c_3,z_{1,L},z_2,z_3) \leftarrow \sigma$
$h \leftarrow H(\rho,m)$
$\forall j \in L,\ T_{1,j} \leftarrow e(I, {}^{z_{1,j}} y_j)\ {}^{c_{1,j}} V$
$T_2 \leftarrow e(z_2,g)\ {}^{c_2} V,\ T_2' \leftarrow e(z_2,h)\ {}^{c_2} U$
$T_3 \leftarrow e(I,z_3)\ {}^{c_3} U$
$c_1 = \sum_{j\in L} c_{1,j}$

$$\text{return } \begin{cases} 1 & \text{if } H'(\rho,L,m,y_L,U,V,T_{1,L},T_2,T_2',T_3) \\ & \quad = (c_1,c_2,c_3), \\ 0 & \text{otherwise.} \end{cases}$$

4.4 提案

エージェント指定匿名化可能署名 [9] においては, まずエージェントが公開パラメタを決めてから署名者は公開鍵

を作らなければならない. 従って, 公開鍵を作った後で他のエージェントを指定したくなった場合公開鍵を作り直す必要がある. 前述の方式に関してはエージェントが生成元 g を生成し, それを用いて署名者が署名を作成することが問題であった. 本稿では, 上記課題を解決する為, 共通参照文字列 crs がシステムパラメタとして事前に決定され, そのハッシュ値によって誰も離散対数を知らない生成元 g を構成するよう上記のエージェント指定匿名化可能署名を改良する.

従って改良された署名方式 Σ は次の構文を満たす 6 つの確率的多項式時間アルゴリズム ($\mathtt{CrsGen}, \mathtt{Setup}, \mathtt{KeyGen}, \mathtt{Sign}, \mathtt{Anonymize}, \mathtt{Verify}$) により構成される.

共通参照文字列生成アルゴリズム $\mathtt{CrsGen}(1^k) \xrightarrow{\$} \rho$: は安全パラメタ 1^k を入力としシステムパラメタ ρ を出力とする確率的多項式時間アルゴリズム.

エージェント鍵生成アルゴリズム $\mathtt{Setup}(\rho) \xrightarrow{\$} (w, g_A)$: はシステムパラメタ ρ を入力としエージェント秘密鍵 w およびエージェント公開鍵 g_A を出力とする確率的多項式時間アルゴリズム.

鍵生成アルゴリズム $\mathtt{KeyGen}(\rho) \xrightarrow{\$} (x, y)$: はシステムパラメタ ρ を入力とし秘密鍵 x および公開鍵 y を出力とする確率的多項式時間アルゴリズム.

署名アルゴリズム $\mathtt{Sign}(x, m, g_A) \xrightarrow{\$} \sigma$: は秘密鍵 x およびメッセージ m エージェント公開鍵 g_A を入力とし署名 σ を出力とする確率的多項式時間アルゴリズム.

匿名化アルゴリズム $\mathtt{Anonymize}(i, \sigma, L, y_L, m) \xrightarrow{\$} \sigma/\perp$: は署名者 ID i, 署名 r, リング $L \subset N$, 公開鍵リスト y_L, およびメッセージ m を入力とし, リング署名 σ' または拒絶 $\perp$ を出力とする確率的多項式時間アルゴリズム.

検証アルゴリズム $\mathtt{Verify}(L, y_L, m, \sigma') \xrightarrow{\$} 0/1$: はリング $L \subset N$, 公開鍵リスト y_L, メッセージ m, およびリング署名 σ', を入力とし, 単一ビット $b \in \{0,1\}$ を出力とする確率的多項式時間アルゴリズム.

構文 : 如何なる (多項式長の) メッセージ $m \in \{0,1\}^*$, 如何なる (多項式長の) リング $L \subset N$, および如何なる署名者 $i \in L$ に対しても

$$\Pr\left[\, b = 0 \;\middle|\; \begin{array}{l} \rho \xleftarrow{\$} \mathtt{CrsGen}(1^k), \\ (w, g_A) \xleftarrow{\$} \mathtt{Setup}(\rho), \\ \forall j \in L, (x_j, y_j) \xleftarrow{\$} \mathtt{KeyGen}(\rho), \\ \sigma \xleftarrow{\$} \mathtt{Sign}(x_i, m, g_A), \\ \sigma' \xleftarrow{\$} \mathtt{Anonymize}(w, \sigma, L, y_L, m), \\ b \xleftarrow{\$} \mathtt{Verify}(L, y_L, m, \sigma'), \end{array} \right]$$

が k に関して無視可能.

署名者の匿名性を壊さないよう署名 σ は署名者とエージェントの間の秘密としなければならないが, σ が攻撃者に漏洩しても, エージェント秘密鍵 w が漏洩していなければ σ が どの x_i によって作成された署名なのかは計算量的に識別困難とする. 本稿ではこのようなエージェント指定匿名化可能署名をエージェント独立エージェント指定匿名化可能署名と呼ぶ.

4.5 具体的構成

$\mathtt{CrsGen}(1^k)$:
  $\mathtt{crs} \leftarrow \{0,1\}^k$
  $g \leftarrow H(\mathtt{crs})$
  $\rho \leftarrow (\mathbb{L}, \mathbb{G}, \mathbb{G}_T, e, g, H, H')$
  return (ρ)

$\mathtt{Setup}(\rho)$:
  $w \xleftarrow{\$} \mathbb{L}^*$
  $g_A \leftarrow I^w$
  return (w, g_A)

$\mathtt{KeyGen}(\rho)$:
  $(\mathbb{L}, \mathbb{G}, \mathbb{G}_T, e, g, H, H') \leftarrow \rho$
  $x \xleftarrow{\$} \mathbb{L},\ y \leftarrow {}^x g$
  return (x, y)

$\mathtt{Sign}(x, m, g_A)$:
  return $\sigma \leftarrow ({}^x H(\rho, m),\ {}^x g_A)$

$\mathtt{Anonymize}(i, \sigma, L, y_L, m)$:
  $h \leftarrow H(\rho, m)$
  $(r, Y) \leftarrow \sigma$
  $X \leftarrow Y^{w^{-1}}$
  if $e(X, g) \neq e(I, y_i)$ return $\perp$
  if $e(I, r) \neq e(X, h)$ return $\perp$

  $(\exists j \in L, e(I, r) = e(X, h))$ なる r, X の知識証明を次のように生成する:
    $t_1, t_2, t_3, \beta \xleftarrow{\$} \mathbb{L}$
    $V \leftarrow {}^{\beta} e(I, y_i),\ U \leftarrow {}^{\beta} e(I, r)$
    $T_{1,i} \leftarrow {}^{t_1} e(I, y_i)$
    $\forall j \in L \setminus \{i\}$
      $c_{1,j}, z_{1,j} \xleftarrow{\$} \mathbb{L}$
      $T_{1,j} \leftarrow e(I, {}^{z_{1,j}} y_j)\ {}^{c_{1,j}} V$
    $T_2 \leftarrow {}^{t_2} e(I, g),\ T_2' \leftarrow {}^{t_2} e(I, h)$
    $T_3 \leftarrow {}^{t_3} e(I, I)$

$$
\begin{aligned}
&(c_{1,i}, c_2, c_3) \leftarrow \\
&\quad H'(\rho, L, m, y_L, U, V, T_{1,L}, T_2, T_2', T_3) \\
&\qquad - \textstyle\sum_{j \neq i} (c_{1,j}, 0, 0) \\
&z_{1,i} \leftarrow t_1 - c_{1,i}\beta \\
&z_2 \leftarrow {}^{t_2}I/^{c_2\beta}X \\
&z_3 \leftarrow {}^{t_3}I/^{c_3\beta}r
\end{aligned}
$$

return $\sigma' \leftarrow (U, V, c_{1,L}, c_2, c_3, z_{1,L}, z_2, z_3)$

$\texttt{Verify}(L, y_L, m, \sigma')$:

$(U, V, c_{1,L}, c_2, c_3, z_{1,L}, z_2, z_3) \leftarrow \sigma'$

$h \leftarrow H(\rho, m)$

$\forall j \in L,\ \ T_{1,j} \leftarrow e(I, {}^{z_{1,j}}y_j)\ \ {}^{c_{1,j}}V$

$T_2 \leftarrow e(z_2, g)\ \ {}^{c_2}V,\ \ T_2' \leftarrow e(z_2, h)\ \ {}^{c_2}U$

$T_3 \leftarrow e(I, z_3)\ \ {}^{c_3}U$

$c_1 = \sum_{j \in L} c_{1,j}$

$$
\text{return } \begin{cases} 1 & \text{if } H'(\rho, L, m, y_L, U, V, T_{1,L}, T_2, T_2', T_3) \\ & \quad = (c_1, c_2, c_3), \\ 0 & \text{otherwise.} \end{cases}
$$

参考文献

[1] I. M. Duursma and N. Kiyavash. The vector decomposition problem for elliptic and hyperelliptic curves. *IACR Cryptology ePrint Archive*, 2005:31, 2005.

[2] I. M. Duursma and S. Park. Elgamal type signature schemes for n-dimensional vector spaces. *IACR Cryptology ePrint Archive*, 2006:312, 2006.

[3] A. Escala, G. Herold, E. Kiltz, C. Ràfols, and J. L. Villar. An algebraic framework for diffie-hellman assumptions. In R. Canetti and J. A. Garay, editors, *Advances in Cryptology - CRYPTO 2013 - 33rd Annual Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2013. Proceedings, Part II*, volume 8043 of *Lecture Notes in Computer Science*, pages 129–147. Springer, 2013.

[4] S. D. Galbraith, K. G. Paterson, and N. P. Smart. Pairings for cryptographers. *Discrete Applied Mathematics*, 156(16):3113–3121, 2008.

[5] S. D. Galbraith and E. R. Verheul. An analysis of the vector decomposition problem. In R. Cramer, editor, *Public Key Cryptography*, volume 4939 of *Lecture Notes in Computer Science*, pages 308–327. Springer, 2008.

[6] J. Groth and A. Sahai. Efficient noninteractive proof systems for bilinear groups. *SIAM J. Comput.*, 41(5):1193–1232, 2012.

[7] F. Hoshino. A Variant of Diffie-Hellman Problem and How to Prove Independency. In *Proc. of SCIS 2014 2014 Symposium on Cryptography and Information Security 2014*. IEICE, 2014.

[8] F. Hoshino, T. Kobayashi, and K. Suzuki. Anonymizable Signature and Its Construction from Pairings. In M. Joye, A. Miyaji, and A. Otsuka, editors, *Pairing-Based Cryptography - Pairing 2010 - 4th International Conference, Yamanaka Hot Spring, Japan, December 2010. Proceedings*, volume 6487 of *Lecture Notes in Computer Science*, pages 62–77. Springer, 2010.

[9] T. Kobayashi and F. Hoshino. Designated Agent Anonymizable Signature. In *Proc. of SCIS 2019 2019 Symposium on Cryptography and Information Security 2019*. IEICE, 2019.

[10] T. Okamoto and K. Takashima. Homomorphic encryption and signatures from vector decomposition. In S. D. Galbraith and K. G. Paterson, editors, *Pairing*, volume 5209 of *Lecture Notes in Computer Science*, pages 57–74. Springer, 2008.

[11] T. Okamoto and K. Takashima. Fully secure functional encryption with general relations from the decisional linear assumption. In T. Rabin, editor, *CRYPTO*, volume 6223 of *Lecture Notes in Computer Science*, pages 191–208. Springer, 2010.

[12] T. Okamoto and K. Takashima. Decentralized attribute-based signatures. *IACR Cryptology ePrint Archive*, 2011:701, 2011.

[13] T. Okamoto and K. Takashima. Fully secure unbounded inner-product and attribute-based encryption. *IACR Cryptology ePrint Archive*, 2012:671, 2012.

[14] H. SHIZUYA and T. TAKAGI. A Public-Key Cryptosystem Based upon Generalized Inverse Matrix over Discrete Logarithmic Domain of Finite Field. **電子情報通信学会論文誌** *A*, J71-A(3):825–832, 1988.

[15] F. Vercauteren, editor. Final Report on Main Computational Assumptions in Cryptography. ECRYPT II European Network of Excellence in Cryptology II, Deliverables of Multi-party and Asymmetric Algorithms Virtual Lab. (MAYA), D.MAYA.6, 2013.

[16] M. Yoshida, S. Mitsunari, and T. Fujiwara. Vector Decomposition Problem and the Trapdoor Inseparable Multiplex Transmission Scheme based the Problem. In *Proc. of SCIS 2003 The 2003 Symposium on Cryptography and Information Security Hamamatsu, Japan, Jan. 26-29, 2003*. IEICE, 2003.