博 士 論 文

# 電子商取引の高セキュリティ化に関する研究

指導教官　宮地 充子 助教授

北陸先端科学技術大学院大学
情報科学研究科情報システム学専攻

田村 裕子

2004 年 3 月

## 要 旨

高度情報化・ネットワークの進展に伴い，多種多様な電子商取引が普及しつつある．電子商取引の普及にともない，悪意あるユーザによる情報の改ざん，盗聴などの犯罪が深刻化している．本稿では，このような問題を防ぐ手法として，犯罪の未然防止を目的とした電子保証人方式，情報の漏洩防止を目的とした匿名証明書方式，低セキュリティなアプリケーションの再構築等による電子商取引技術の高セキュリティ化をおこなう．

# 目 次

# 第 1章

# まえがき

高度情報化・ネットワークの進展に伴い，電子オークションに代表される多種多様な電子商取引が普及しつつある反面，悪意ある取引相手による情報の改ざん，漏洩などの犯罪が深刻化している．このような問題を防ぐ方法として，犯罪の未然防止，情報の漏洩防止，低セキュリティなアプリケーションの再構築による電子商取引技術の高セキュリティ化が考えられる．未然防止には，取引にまつわる情報が信頼できるものであるか判断できる技術が必要であり，漏洩防止には，取引の際の個人情報が，取引相手 (機関) 以外に漏洩しない仕組みが有効である．

具体的手法として，犯罪の未然防止を目的とした第三者による保証という概念を実現する電子保証人方式，情報の漏洩防止を目的とした匿名証明書方式を提案する．また，複数のユーザが同時に参加する取引形態である，eBay，Yahoo[2, 1] に代表されるオークションにセキュリティ技術を導入したシステムの再構築をおこなう．

**電子保証人方式に関する研究**

電子商取引はインターネットの発達により，その利便性から近年広く普及している．その一方で，あらゆるユーザの参加を可能とする電子商取引にまつわる犯罪の増加はインターネットの裾野が広がるにつれ深刻化しており，見知らぬユーザとの取引はリスクを伴うものとなっている．このような犯罪を未然に防ぐには，取引内容の信用性，取引相手 (機関) の信頼性の確保が必要不可欠である．

このような取引内容，取引相手等の情報の正当性を保証するには，実社会で

利用される信頼できる第三者による保証という慣習が有効である．そこで本研究では，保証という概念を電子的に実現することを目的とする．取引の電子保証の形態は，取引内容の保証と取引相手自身の保証の 2 形態に分類でき，前者は取引相手 (機関) が取引内容に対して信頼できる第三者の保証を受けることを意味し，後者は取引相手がおこなう全ての取引が第三者によって保証されることを意味する．このような電子保証人方式において，検証者はいづれの場合も与えられた文書が該当取引相手により生成されたものであり，かつ第三者により保証されているということを確認できる必要がある．

保証者とは，契約者 (取引相手) が契約違反を犯したとき，代わりに契約を遂行するユーザであり，保証者の役割は契約者の役割を包含する．よって，電子保証人方式にて，保証者が契約者の上位にあるという保証概念を電子的に実現する．そこで，1. 保証者と契約者の確認ができる，2. 契約者の確認はおこなわず，保証者のみの確認をおこなうことができる，3. 保証者の確認はおこなわず，契約者のみの確認をおこなうことができない，という性質をもたせることによって，契約者と保証者の役割の相違を電子保証人方式として反映させる．本稿では，電子保証方式の満たすべき性質を定義し，それを実現する手法を提案する．

**匿名証明書方式に関する研究**

ユーザが病院，クレジットカード会社，商店等の各機関と取引をおこなう場合，各機関はユーザに関する情報を保有することになる．仮に，各機関が取引のあるユーザの個人情報を流出した場合，普及したインターネット網はまたたく間に個人情報を拡散することになるだろう．それによって，各機関が独自に所有していたユーザの情報が結合し，容易に個人に関わる全ての情報が漏洩することになる．このような情報の漏洩・拡大から個人情報を守るには，それらを扱う機関の結託も考慮した技術の提供が必要であり，個人ユーザが各機関に提供する情報の利用方法を自ら制御できる必要がある．

Chaum によって提案された匿名証明書方式 (Anonymous Credential System)[15] は，ユーザと機関の間の個人情報保護を実現する．ユーザは各機関に異なる仮名 (Pseudonym) で登録し，機関はその仮名によってユーザの情報を管理・

識別する．異なる仮名は各機関の所有する個人情報の関連付けを不可能にし，機関の結託による情報の拡大を防ぐことができる．また，各機関は登録者であることを保証する登録証明書をユーザに発行し，ユーザは別の機関 (検証者) に発行された証明書を提示することで登録者であることを証明できる．この証明書の提示 (保持証明) は証明書の発行機関へ登録済みであるという事実以外，ユーザに関する情報を何も漏らさない．つまり，仮名と本人との結びつきを防ぐことで，機関が結託してもユーザの個人情報が漏洩しない．

しかしながら，既存の方式では登録証明に必要な情報をユーザが制御できない．このため，登録証明書の保持事実がユーザの個人情報を不必要に漏らす可能性がある．例えば，学校への登録という状況について考える．在学を証明する学生証は学校 (長) によって発行された登録証明書であり，当該大学の学生であることを保証し，学生証の提示 (保持証明) は検証者に所属学校名を与えることになる．しかし，学割サービス等のように学生である事実を証明できればよい場合，学生証の提示は過度な情報を与えるだろう．一方，当該大学の学生のみが享受できるサービスには，検証者は証明書の発行元 (学校) の情報が必要である．つまり，フレキシブルな個人情報の管理には，単に学生であることと，当該学校に所属していることを用途に応じて証明できることが望ましい．

一般に，検証者によってユーザが当該機関に属していることが要求される場合，その事実を証明する必要がある．しかしながら，検証者が個々の機関に登録しているかどうかではなく，機関を束ねるグループに属しているかを要求するとき，ユーザは当該機関への登録という不要な情報を示す必要はない．そこで，本稿ではユーザにデータ管理の負荷を与えず，フレキシブルな個人情報の保護を可能とする匿名証明書方式を構築する．本提案では，検証者のセキュリティポリシーに応じて，機関の登録証明書の保持，グループ内のある機関の登録証明書の保持というように検証者に与える情報をユーザ自身で制御することができる．

一方，相違なる複数の機関の登録証明の対応という問題もある．例えば，学割の海外旅行ツアーに参加する場合には，パスポートと学生証という 2 つの

登録証明書の保持証明が必要である．既存の方法では，1 機関への登録に対して 1 つの証明書が発行されているため，$m$ 個の登録証明が求められた場合，ユーザは独立な $m$ 個の証明書の保持を証明しなければならない．また，一つの機関に複数の登録保証権限を与えると，その証明書から登録しているすべての機関が露見することになる．本稿では，ユーザの全ての登録機関の中から任意の複数機関への登録のみを効率的に示すことのできる複数匿名証明書方式を実現する．

**電子オークションに関する研究**

複数ユーザが一度に参加する取引形態であるイングリッシュ・オークションは，インターネット上においても広く普及し，その参加者は急激に増加している．イングリッシュ・オークションとは，公開される現在の落札価格より高い値を順に入札していき，オークション開催時間内での最高入札額が落札額となり，その入札者が落札者となる価格吊り上げ型のオークションである．インターネット上で開催されるイングリッシュ・オークションにおいて商品を落札するためには，参加者は他の入札値を随時チェックして入札を繰り返す必要がある．そのような欠点を克服した方式が代理入札 (Proxy-bidding) である．代理入札では，各参加者は希望落札額を代理人に提示し，代理人が参加者に代わって入札をおこなうため，落札者決定まで，参加者はインターネットに束縛される必要がないという利点をもつ．実際，インターネット上で広く普及しているオークションはこの代理入札であり [2, 1]，代理人の役割をオークション管理者が一括して担うことで実現されている．この場合，オークション管理者は新たな入札 (落札希望額の提示) がおこなわれたとき，その時点での落札者の落札希望額と新たな入札者の落札希望額を比較し，低額 (+入札幅) を現在の落札価格とする．

オークションでは，入札値情報は市場価格としての価値をもつ．したがって，安全な電子オークションには，ユーザの入札値の秘匿性，入札値からユーザを特定できない匿名性，オークションが正しく行われていることを検証できる公開検証性が必要である．さらに，代理入札方式では，価格更新前までの入札値の秘匿性と，価格更新後の落札者の入札値の秘匿性，1 回の入札の効率

性，価格更新の効率性も必要となる．しかしながら，現在普及する電子オークションは入札値の秘匿性や公開検証性をみたしていない．

塩月・宮地は上記の性質をみたす代理入札システム [52] を提案したが，効率的とは言い難い．そこで本稿では，塩月らの方式のもつ安全性を損なわず，より効率的な手法の提案を目的とする．

# 第 2章

# 準備

本章では，公開鍵暗号系に必要な問題，及び仮定を列記する．

## 2.1 数論的問題

**定義 1** (Diffie-Hellman 問題) 有限群 $G$，元 $g \in G$ と 2 つの元 $g^x, g^y \in G$ が与えられたとき，$g^{xy}$ を求める問題を Diffie-Hellman 問題という．

**仮定 1** (Diffie-Hellman 問題仮定) 有限群 $G$，元 $g \in G$ と 2 つの元 $g^x, g^y \in G$ の入力に対し，無視できない確率で $g^{xy}$ を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．

**定義 2** (Diffie-Hellman 決定問題) 有限群 $G$，元 $g \in G$ と 3 つの元 $g^x, g^y, g^w \in G$ が与えられたとき，$g^{xy} = g^w$ かどうかを判定する問題を Diffie-Hellman 決定問題という．

**仮定 2** (Diffie-Hellman 決定問題仮定) 有限群 $G$，元 $g \in G$ と 3 つの元 $g^x, g^y, g^w \in G$ の入力に対し，無視できない確率で $g^{xy} = g^w$ かどうか判定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．

**定義 3** (離散対数問題) 有限群 $G$，元 $g, y \in G$ が与えられたとき，$y = g^x$ をみたす元 $0 \leq x \leq |G|$ を求める問題を離散対数問題という．

**仮定 3** (離散対数問題仮定) 有限群 $G$，元 $g, y \in G$ の入力に対し，無視できない確率で $y = g^x$ をみたす元 $0 \leq x \leq |G|$ を出力する確率的多項式時間アルゴリズムは存在しない.

**定義 4** (素因数分解問題) ある数 $n \in \mathbb{Z}$ が与えられたとき，$n = {p_1}^{e_1}{p_2}^{e_2} \cdots {p_k}^{e_k}$ となる互いに異なる素数 $p_i$ $(1 \leq i \leq k)$ と正整数 $e_i \in \mathbb{Z}_{>0}$ を求める問題を素因数分解問題という.

**仮定 4** (素因数分解問題仮定) ある数 $n \in \mathbb{Z}$ の入力に対し，無視できない確率で $n = {p_1}^{e_1}{p_2}^{e_2} \cdots {p_k}^{e_k}$ となる互いに異なる素数 $p_i$ $(1 \leq i \leq k)$ と正整数 $e_i \in \mathbb{Z}_{>0}$ を出力する確率的多項式時間アルゴリズムは存在しない.

**定義 5** ($e$ 乗根問題) 合成数 $n = pq, z \in \mathbb{Z}_n^*$ および $e \in \mathbb{Z}_{>1}$ が与えられたとき，$z \equiv u^e \pmod{n}$ をみたす $u$ を求める問題を $e$ 乗根問題という.

**仮定 5** ($e$ 乗根問題仮定) 合成数 $n = pq, z \in \mathbb{Z}_n^*$ および $e \in \mathbb{Z}_{>1}$ の入力に対し，無視できない確率で $z \equiv u^e \pmod{n}$ をみたす $u$ を出力する確率的多項式時間アルゴリズムは存在しない.

**定義 6** (強 RSA 問題) 合成数 $n = pq, z \in \mathbb{Z}_n^*$ が与えられたとき，$z \equiv u^e \pmod{n}$ をみたす $u \in \mathbb{Z}_n^*$ と $e \in \mathbb{Z}_{>1}$ を求める問題を強 RSA 問題という.

**仮定 6** (強 RSA 問題仮定) 合成数 $n = pq, z \in \mathbb{Z}_n^*$ の入力に対し，無視できない確率で $z \equiv u^e \pmod{n}$ をみたす $u \in \mathbb{Z}_n^*$ と $e \in \mathbb{Z}_{>1}$ を出力する確率的多項式時間アルゴリズムは存在しない.

**定義 7** どのような正定数 $c$ に対しても，十分大きな $n$ に対して

$$\forall c \exists N \forall n \ [n > N \rightarrow f(n) < 1/n^c]$$

であるとき，$f(n)$ は無視できるという.

**定義 8** どのような正定数 $c$ に対しても，十分大きな $n$ に対して

$$\forall c \exists N \forall n [n > N \rightarrow f(n) > 1 - 1/n^c]$$

であるとき，$f(n)$ は圧倒的であるという.

## 2.2 公開鍵暗号

公開鍵暗号とは，平文を暗号化するための鍵と，復号するための鍵が異なる非対称暗号方式であり，暗号鍵を公開することができる暗号方式である．

データの送信者は相手の公開鍵を用いて平文を暗号化し，公開鍵に対応する秘密鍵を保持するユーザのみが暗号文を復号することができる．

### 2.2.1 RSA 暗号

RSA 暗号 [48] は Rivest, Shamir, Adleman によって提案された初めての公開鍵暗号であり，素因数分解問題の困難さを安全性の根拠とした手法である．

1. 鍵生成: ユーザは，素数 $p, q$ を生成し，$n = pq, \lambda(n) = \mathrm{lcm}(p-1, q-1)$ に対して，素数 $e \in_R \mathbb{Z}_{\lambda(n)}$ を選択し，$ed \equiv 1 \pmod{\lambda(n)}$ をみたす $d$ を求める．ユーザは $(p, q, d)$ を秘密鍵，$(n, e)$ をそれに対応する公開鍵とする．

2. 暗号化: データの送信者は，平文 $m \in \mathbb{Z}_n$ に対し，ユーザの公開鍵を用いて $c := m^e \bmod n$ を計算し，$c$ を暗号文とする．

3. 復号: ユーザは，秘密鍵を用いて $m := c^d \bmod n$ を計算することによって，平文 $m$ を復号する．

### 2.2.2 ElGamal 暗号

ElGamal 暗号 [22] は離散対数問題の困難さを安全性の根拠とした手法であり，準同型性をもつことからマルチ・パーティ・プロトコルなどに広く用いられている．

1. 鍵生成: ユーザは，十分大きな素数 $p$ を生成し，位数を $p-1$ とする元 $g \in \mathbb{Z}_p^*$ を選択する．さらに，秘密鍵 $x \in_R \mathbb{Z}_{p-1}$ を生成し，$(p, g, y := g^x \bmod p)$ をそれに対応する公開鍵とする．

2. 暗号化: データの送信者は，乱数 $r \in_R \mathbb{Z}_{p-1}$ を選択し，平文 $m \in \mathbb{Z}_p^*$ に対し，公開鍵を用いて $c_1 := g^r \bmod p, c_2 := my^r \bmod p$ を計算し，$(c_1, c_2)$ を暗号文とする．

3. 復号: ユーザは，秘密鍵を用いて $m := c_2/{c_1}^x \bmod p$ を計算することによって，平文 $m$ を復号する．

### 2.2.3 Cramer-Shoup 暗号

Cramer-Shoup 暗号 [19] は，Diffie-Hellman 決定問題仮定とハッシュ関数が理想的かつ仮想的なランダム関数であるという仮定のもと，安全であることが知られている．これは ElGamal 暗号の安全性を強化した方式の一つであり，頑強性をもたせるため，復号時に不当な暗号文を棄却するための検証をおこなう．

1. 鍵生成: ユーザは，十分大きな素数 $p$ を生成し，位数を $p-1$ とする元 $g_1, g_2 \in \mathbb{Z}_p^*$ とハッシュ関数 $\mathcal{H} : \{0,1\}^* \to \{0,1\}^{|p-1|}$ を設定する．さらに秘密鍵 $x_1, x_2, x_3, x_4, x_5 \in_R \mathbb{Z}_{p-1}$ を生成し，$(p, g_1, g_2, y_1 := {g_1}^{x_1}{g_2}^{x_2} \bmod p, y_2 := {g_1}^{x_3}{g_2}^{x_4} \bmod p, y_3 := {g_1}^{x_5} \bmod p)$ をそれに対応する公開鍵とする．

2. 暗号化: データの送信者は，乱数 $r \in_R \mathbb{Z}_{p-1}$ を選択する．さらに，平文 $m \in \mathbb{Z}_p$ に対し，公開鍵を用いて $c_1 := {g_1}^r \bmod p, c_2 := {g_2}^r \bmod p, c_3 := m{y_3}^r \bmod p$ と，$e := \mathcal{H}(c_1, c_2, c_3, m)$ に対する $c_4 := (y_1{y_2}^e)^r \bmod p$ を計算し，$(c_1, c_2, c_3, c_4)$ を暗号文とする．

3. 復号: ユーザは，$\tilde{e} := \mathcal{H}(c_1, c_2, c_3, m)$ とし，秘密鍵を用いて $c_4 = {c_1}^{x_1+x_3\tilde{e}}{c_2}^{x_2+x_4\tilde{e}} \bmod p$ が成り立つならば $m = c_3/{c_1}^{x_5} \bmod p$ を計算することによって，平文 $m$ を復号する．

## 2.3 ディジタル署名

ディジタル署名とは，公開鍵の応用によってディジタル文書の正当性を保証する技術であり，署名者は秘密鍵を用いてメッセージに署名をおこない，検証者は署名者の公開鍵を用いてその事実を確認することができる．

### 2.3.1 RSA 署名

RSA 署名は素因数分解問題の困難さを安全性の根拠としたディジタル署名方式である．RSA 署名は用いるハッシュ関数が理想的かつ仮想的なランダム関数であると仮定すると，$e$ 乗根問題仮定のもと，安全であることが証明されている [9].

1. 鍵生成: 署名者は，素数 $p, q$ を生成し，$n = pq, \lambda(n) = \mathrm{lcm}(p-1, q-1)$ に対して，素数 $e \in_R \mathbb{Z}_{\lambda(n)}$ を選択し，$ed \equiv 1 \pmod{\lambda(n)}$ をみたす $d$ を求める．また，ハッシュ関数 $\mathcal{H} : \{0,1\}^* \to \{0,1\}^{|n|}$ を用意し，$(p, q, d)$ を秘密鍵，$(n, e)$ をそれに対応する公開鍵とする．

2. 署名生成: 署名者は，メッセージ $m$ に対して，秘密鍵を用いて $\sigma = \mathcal{H}(m)^d \bmod n$ を計算し，$\sigma$ を $m$ に対する署名とする．

3. 署名検証: 検証者は，メッセージ $m$ に対する $\mathcal{H}(m)$ を計算し，公開鍵に対して，$\mathcal{H}(m) \equiv \sigma^e \pmod{n}$ が成り立つならば署名を受理し，そうでないならば拒否する．

### 2.3.2 Schnorr 署名

Schnorr 署名 [47] は離散対数問題の困難さを安全性の根拠とした，3 交信認証に基づくディジタル署名である．ハッシュ関数が理想的，かつ仮想的なランダム関数であると仮定したとき安全であることが証明されている [46].

1. 鍵生成: 署名者は，$q \mid p-1$ をみたす十分大きな素数 $p, q$ を生成し，$q$ を位数とする元 $g \in \mathbb{Z}_p^*$ とハッシュ関数 $\mathcal{H} : \{0,1\}^* \to \{0,1\}^{|q|}$ を設定する．さらに，秘密鍵 $x \in_R \mathbb{Z}_q^*$ を生成し，$(p, q, g, y := g^x \bmod p)$ をそれに対応する公開鍵とする．

2. 署名生成: 署名者は，乱数 $r \in_R \mathbb{Z}_q^*$ を選択し，$t := g^r \bmod p$ を計算する．次に，秘密鍵を用いてメッセージ $m$ に対する $e := \mathcal{H}(m, t)$ を求め，$s := r - ex \bmod q$ とし，$(e, s)$ をメッセージ $m$ に対する署名とする．

3. 署名検証: 検証者は，公開鍵を用いて $\tilde{t} := g^s y^e \bmod p$ を計算し，$e = \mathcal{H}(m, \tilde{t})$ が成り立つならば署名を受理し，そうでないならば拒否する．

## 2.4 ブラインドディジタル署名

電子現金や電子投票への応用を目的として提案されたブラインド署名は，メッセージの内容を秘密にしたまま，それに対応する署名の生成を依頼する手法であり，依頼者はメッセージ $m$ の内容を署名者に秘密にしたまま，$m$ に対する署名を得ることができる．

### 2.4.1 RSA ブラインド署名

RSA 署名に基づくブラインド署名 [15] を示す．

1. 鍵生成: 署名者は，素数 $p, q$ を生成し，$n := pq, \lambda(n) = \mathrm{lcm}(p-1, q-1)$ に対して，素数 $e \in \mathbb{Z}_{\lambda(n)}$ を選択し，$ed \equiv 1 \pmod{\lambda(n)}$ をみたす $d$ を求める．また，ハッシュ関数 $\mathcal{H} : \{0,1\}^* \to \{0,1\}^{|n|}$ を用意し，$(p, q, d)$ を秘密鍵，$(n, e)$ をそれに対応する公開鍵とする．

2. 署名生成:

   *Step 1.* 依頼者は，乱数 $r \in_R \mathbb{Z}_n^*$ を選択し，公開鍵を用いてメッセージ $m$ に対する $t := r^e \mathcal{H}(m) \bmod n$ を生成し，署名者に送る．

   *Step 2.* 署名者は，秘密鍵を用いて $s := t^d \bmod n$ を生成し，依頼者に送る．

   *Step 3.* 依頼者は，$\sigma := s/r \bmod n$ を計算し，$\sigma$ を $m$ に対する署名とする．

3. 署名検証: 検証者は，メッセージ $m$ に対する $\mathcal{H}(m)$ を計算し，署名者の公開鍵に対して，$\mathcal{H}(m) \equiv \sigma^e \pmod{n}$ が成り立つならば署名を受理し，そうでないならば拒否する．

### 2.4.2 Schnorr ブラインド署名

3 交信認証に基づく署名方式においてもブラインド署名を構成することができる [40]．ここでは，Schnorr 署名に基づくブラインド署名の実現法を示す．

1. 鍵生成: 署名者は，$q \mid p-1$ をみたす十分大きな素数 $p, q$ を生成し，$q$ を位数とする元 $g \in \mathbb{Z}_p^*$ とハッシュ関数 $\mathcal{H} : \{0,1\}^* \to \{0,1\}^{|q|}$ を設定する．さらに，

秘密鍵 $x \in_R \mathbb{Z}_q^*$ を生成し，$(p, q, g, y := g^x \bmod p)$ をそれに対応する公開鍵とする．

2. 署名生成:

*Step 1.* 署名者は，乱数 $r_S \in_R \mathbb{Z}_q^*$ を選択し，$t_S := g^{r_S} \bmod p$ を依頼者に送る．

*Step 2.* 依頼者は，乱数 $r_1, r_2 \in_R \mathbb{Z}_q^*$ を選択し，$t_C := t_S g^{r_1} y^{r_2} \bmod p$ を計算する．次に，メッセージ $m$ に対して $e_C := \mathcal{H}(m, t_C)$ を求め，$e_S := e_C - r_2 \bmod q$ とし，$e_S$ を署名者に送る．

*Step 3.* 署名者は，秘密鍵を用いて $s_S = r_S - e_S x \bmod q$ を計算し，依頼者に送る．

*Step 4.* 依頼者は，$s_C := s_S + r_1 \bmod q$ を求め，$(e_C, s_C)$ をメッセージ $m$ に対する署名とする．

3. 署名検証: 検証者は，署名者の公開鍵に対して $\tilde{t} := g^{s_C} y^{e_C} \bmod p$ を計算し，$e_C = \mathcal{H}(m, \tilde{t})$ がなりたつならば署名を受理し，そうでないならば拒否する．

## 2.5 対話証明

証明者 $P$ が検証者 $V$ に対話をおこなうことによって，ある事実を納得させるプロトコルを考える．

**定義 9** $P$ を確率的チューリングマシン，$V$ を多項式時間確率的チューリングマシンとしたとき，言語 $L$ に対して，

**完全性:** どのような正定数 $c$ に対しても，十分大きな $x \in L$ に対して

$$\forall c \exists N \forall x [|x| > N \rightarrow Pr[P, V \text{の組が} x \text{を受理}] > 1 - 1/|x|^c]$$

**健全性:** どのような正定数 $c$，確率的チューリングマシン $P^*$ に対しても，十分大きな $x \notin L$ に対して

$$\forall c \exists N \forall x [|x| > N \rightarrow Pr[P^*, V \text{の組が} x \text{を受理}] < 1/|x|^c]$$

をみたすとき，$P, V$ の組は言語 $L$ に対する対話証明であるという．

対話証明における $P$ と $V$ との対話をシミュレートするようなシミュレータ $S$ を構築することができる場合，証明者 $P$ の秘密に関する情報は漏れていないと考えることができる．このようなシミュレータが構築できるとき，その対話証明はゼロ知識対話証明とよばれ，シミュレートの精度によって，これらは完全ゼロ知識，統計的ゼロ知識，計算量的ゼロ知識の 3 種類に分類することができる．ここで，統計的ゼロ知識に関する定義を与えておく．

**定義 10** $L$ を言語，$\{U(x)\}, \{V(x)\}$ を確率変数の族としたとき，どのような正定数 $c$ に対しても，十分大きな $x \in L$ に対して

$$\forall c \exists N \forall x [|x| > N \rightarrow \sum_{\alpha \in \{0,1\}^*} |Pr[U(x) = \alpha] - Pr[V(x) = \alpha]| < 1/|x|^c]$$

であるとき，$\{U(x)\}$ と $\{V(x)\}$ は $L$ に関して統計的に識別不可能であるという．

このとき，統計的ゼロ知識対話証明は次のように定義される．

**定義 11** $L$ を言語とし，$P, V$ の組をを言語 $L$ に対する対話証明とする．このとき，$View_{(P,V)}(x, y)$ は $P$ と $V$ の対話において $V$ が知ることのできる全ての系列 (共通入力 $x$，$V$ の生成した乱数，$P$ からの対話，その他の外部入力 $y \in \{0,1\}^{|x|^c}$ ($c$:定数)) からなる確率変数の族とする．

全ての $x \in L, y \in \{0,1\}^{|x|^c}$ と全ての多項式時間確率的チューリングマシン $V^*$ に対して，平均的多項式時間チューリングマシン $M_{V^*}$ が存在し，$View_{(P,V^*)}$ と $M_{V^*}(x, y)$ が $L' = \{x, y\}$ に関して統計的に識別不可能であるとき，$P, V$ を統計的ゼロ知識証明であるという．

## 2.6 認証プロトコル

ゼロ知識証明を用いて，公開情報 (公開鍵) に対する秘密情報を明かすことなく，その保持を証明することによってユーザ認証をおこなうことができる．

### 2.6.1 Schnorr 認証プロトコル

認証プロトコルには対話の回数が 3 回である 3 交信認証とよばれる方式がいくつか提案されている．Schnorr 認証プロトコルは離散対数問題の困難さに安全性の

根拠をおく方式であり，Damgard による Auziliary string モデル [21] の上で統計的ゼロ知識証明であることが示されている．

1. 鍵生成: 証明者は，$q \mid p-1$ をみたす十分大きな素数 $p, q$ を生成し，$q$ を位数とする元 $g \in \mathbb{Z}_p^*$ を設定する．さらに，秘密鍵 $x \in_R \mathbb{Z}_q^*$ を生成し，$(p, q, g, y := g^x \bmod p)$ をそれに対応する公開鍵とする．

2. 知識証明:

   *Step 1.* 証明者は，乱数 $r \in_R \mathbb{Z}_q^*$ を選択し，$t := g^r \bmod p$ を，検証者に送る．

   *Step 2.* 検証者は，$e \in_R \{0,1\}^{|q|}$ を選択し，証明者に送る．

   *Step 3.* 証明者は，秘密鍵を用いて $s = r - ex \bmod q$ を計算し，検証者に送る．

   *Step 4.* 検証者は，$t = g^s y^e \bmod p$ が成り立つとき，証明者が公開鍵 $y, (g, p, q)$ に対応する秘密鍵の保持者であることを確認する．

本稿では以後，この Schnorr の認証プロトコルを $PK\{(\alpha) : y = g^\alpha\}$ と表記し，$y, g \in \mathbb{Z}_p^*$ に対して，$y = g^\alpha \bmod p$ をみたす $\alpha$ の統計的ゼロ知識証明をあらわすことにする．また，一般に

$$PK\{(\alpha_1, \ldots, \alpha_u) : \text{述語}\}$$

で，述語をみたす $\alpha_1, \ldots, \alpha_u$ のゼロ知識対話証明をあらわすことにする．また，Schnorr の認証プロトコルでは演算を $\mathbb{Z}_p$ 上でおこなうが，これを合成数 $n$ に対する群 $\mathbb{Z}_n$ 上に拡張することができる．

ここで，位数を証明者の秘密情報とする群 $G = \langle g \rangle$，但し $|G| = \ell_G, \varepsilon > 1$，セキュリティ・パラメータを $k$ とした場合の例をいくつか挙げておく．

1. $y = g^x$ をみたす $x$ の統計的ゼロ知識証明: $PK\{(\alpha) : y = g^\alpha\}$

   *Step 1.* 証明者は，乱数 $r \in_R \pm\{0,1\}^{\varepsilon(\ell_G + k)}$ を選択し，$t := g^r$ を，検証者に送る．

   *Step 2.* 検証者は，$e \in_R \{0,1\}^k$ を選択し，$e$ を証明者に送る．

   *Step 3.* 証明者は，$s := r - ex \in \mathbb{Z}$ を計算し，$s$ を検証者に送る．

*Step 4.* 検証者は，$t = g^s y^e$ が成り立つとき，証明者が $y = g^\alpha$ をみたす $\alpha$ の保持者であることを確認する．

2. $y = \prod_{i=1}^{u} {g_i}^{x_i}$ をみたす $x_i$ $(1 \leq i \leq u)$ の統計的ゼロ知識証明:
$PK\{(\alpha_1, \ldots, \alpha_u) : y = \prod_{i=1}^{u} {g_i}^{\alpha_i}\}$

*Step 1.* 証明者は，$r_1, \ldots, r_u \in_R \pm\{0,1\}^{\varepsilon(\ell_G+k)}$ を選択し，$t_1 := g^{r_1}, \ldots, t_u := g^{r_u}$ を，検証者に送る．

*Step 2.* 検証者は，$e \in_R \{0,1\}^k$ を選択し，$e$ を証明者に送る．

*Step 3.* 証明者は，$s_1 := r_1 - ex_1 \in \mathbb{Z}, \ldots, s_u := r_u - ex_u \in \mathbb{Z}$ を計算し，$s_i$ $(1 \leq i \leq u)$ を検証者に送る．

*Step 4.* 検証者は，$\prod_{i=1}^{u} t_i = y^e \prod_{i=1}^{u} {g_i}^{s_i}$ が成り立つとき，証明者が $y = \prod_{i=1}^{u} {g_i}^{\alpha_i}$ をみたす $\alpha_i$ $(1 \leq i \leq u)$ の保持者であることを確認する．

3. $y_1 = \prod_{i \in \mathcal{J}_1} {g_i}^{x_{e_{1i}}} \wedge \cdots \wedge y_n = \prod_{i \in \mathcal{J}_n} {g_i}^{x_{e_{ni}}}$ をみたす $x_i$ $(1 \leq i \leq u)$ の統計的ゼロ知識証明: $PK\{(\alpha_1, \ldots, \alpha_u) : y_1 = \prod_{i \in \mathcal{J}_1} {g_i}^{\alpha_{e_{1i}}} \wedge \cdots \wedge y_n = \prod_{i \in \mathcal{J}_n} {g_i}^{\alpha_{e_{ni}}}\}$

*Step 1.* 証明者は，$r_1, \ldots, r_u \in_R \pm\{0,1\}^{\varepsilon(\ell_G+k)}$ を選択し，$t_1 := g^{r_1}, \ldots, t_u := g^{r_u}$ を，検証者に送る．

*Step 2.* 検証者は，$e \in_R \{0,1\}^k$ を選択し，$e$ を証明者に送る．

*Step 3.* 証明者は，$s_1 := r_1 - ex_1 \in \mathbb{Z}, \ldots, s_u := r_u - ex_u \in \mathbb{Z}$ を計算し，$s_i$ $(1 \leq i \leq u)$ を検証者に送る．

*Step 4.* 検証者は，$\prod_{i \in \mathcal{J}_1} t_{c_{1i}} = {y_1}^e \prod_{i \in \mathcal{J}_1} {g_i}^{s_{c_{1i}}}, \ldots, \prod_{i \in \mathcal{J}_n} t_{c_{ni}} = {y_n}^e \prod_{i \in \mathcal{J}_n} {g_i}^{s_{c_{ni}}}$ が成り立つとき，証明者が上式をみたす $\alpha_i$ $(1 \leq i \leq u)$ の保持者であることを確認する．

4. $y = g^x$ かつ $x \in ]X - 2^{\varepsilon(\ell_G+k)}, X + 2^{\varepsilon(\ell_G+k)}[$ をみたす $x$ の統計的ゼロ知識証明:
$PK\{(\alpha) : y = g^\alpha \wedge \alpha \in ]X - 2^{\varepsilon(\ell_G+k)}, X + 2^{\varepsilon(\ell_G+k)}[\}$

*Step 1.* 証明者は，$r \in_R \pm\{0,1\}^{\varepsilon(\ell_G+k)}$ を選択し，$t := g^r$ を，検証者に送る．

*Step 2.* 検証者は，$e \in_R \{0,1\}^k$ を選択し，$e$ を証明者に送る．

*Step 3.* 証明者は，$s := r - e(x - X) \in \mathbb{Z}$ を計算し，$s$ を検証者に送る．

*Step 4.* 検証者は，$t = g^{s-eX}y^e$ が成り立つとき，証明者が $y = g^\alpha$ かつ $\alpha \in ]X - 2^{\varepsilon(\ell+k)}, X + 2^{\varepsilon(\ell+k)}[$ をみたす $\alpha$ の保持者であることを確認する．

また，ハッシュ関数を用いてこのような3交信認証プロトコルに基づくディジタル署名方式を構成することができる．そのような署名を $SPK\{(\alpha_1, \ldots, \alpha_u) : \text{述語}\}(m)$ と記述することにする．これは，述語をみたす $\alpha_1, \ldots, \alpha_u$ の保持者によって生成されたメッセージ $m$ に対する署名を意味する．

本稿では，平方剰余からなる群 $\mathbb{Z}_n^{*2}$ 上で知識証明をおこなう場合がある．この場合，証明者は証明に用いる各要素が $\mathbb{Z}_n^{*2}$ の元であることを検証者に示す必要があるが，これを $PK^2\{(\alpha_1, \ldots, \alpha_u) : \text{述語}\}$ と記述することにする．

## 2.7 暗号プロトコル

本節では，暗号プロトコルとして Shamir の $(t, n)$ 閾値法，分散情報生成プロトコル，暗号化検証，閾値分散復号，分散平文等価テスト，ミックス・アンド・マッチと金持ちの財産比べプロトコル，ビット・スライスプロトコルを紹介する．以下では，$E$ を公開鍵暗号の暗号化関数，$D$ を復号関数，$E_m$ を平文 $m$ の暗号文の集合とする．

### 2.7.1 Shamir の $(t, n)$ 閾値法

秘密鍵などの重要な情報を紛失や盗難から守る手法が秘密分散法である．その中でも Shamir の $(t, n)$ 閾値法 [50] は，秘密情報 $x$ を $n$ 個の分散情報に符号化し，そのうちの任意の $t$ $(\geq n)$ 個以上の分散情報を集めれば $x$ を復元できるが，任意の $t-1$ 個以下の分散情報では $x$ に関する情報を何も得られないという特徴をもつ．

1. 秘密分散: ディーラーは，十分大きな素数 $p$ を生成する．さらに乱数 $a_j \in_R \mathbb{Z}_p$ $(1 \leq j \leq t-1)$ を選択し，高々 $t-1$ 次の多項式 $f(z) = x + \sum_{j=0}^{t-1} a_j z^j \bmod p$ を生成する．プレイヤー $P_i$ への分散情報を $x_i = f(i)$ とし送る．

2. 秘密復元: $t$ 人のプレイヤー $\{P_{i_1}, \ldots, P_{i_t}\}$ はラグランジェの補間公式より $\lambda_j(z) = \prod_{\ell \neq j} z - i_\ell / i_j - i_\ell$ を用いて $f(x) = \sum_{j=1}^{t} \lambda_j(x) f(i_j)$ を求め，$x = f(0)$ を復

元する．

### 2.7.2 分散情報生成プロトコル

Shamir の $(t, n)$ 閾値法では，予め秘密情報を知るディーラーの存在が必要不可欠である．しかし，分散情報生成プロトコルは，そのような信頼できるディーラーを仮定せずに，公開鍵 $y = g^x$ に対応する秘密鍵 $x$ を分散管理する方式である．

秘密鍵 $x$ は，$n$ 人のプレイヤー $P_i$ が選んだ秘密情報 $a_{i0}$ を用いて $x = \sum_{P_i} a_{i0}$ であらわされる．このプロトコルを用いて分散された秘密の復元に用いられる多項式は $P_i$ が定数項を $a_{i0}$ として生成した高々$t-1$ 次多項式 $f_i(z)$ の和，$F(z) = \sum_{P_i} f_i(z)$ となる．

1. 秘密鍵生成: $q \mid p-1$ をみたす十分大きな素数 $p, q$ を，各プレイヤー間で事前に共有しておくものとする．

   *Step 1.* 各プレイヤー $P_i$ は，乱数 $a_{ij}, b_{ij} \in_R \mathbb{Z}_q$ $(0 \leq j \leq t-1)$ を選択し，高々 $t-1$ 次の 2 つの多項式 $f_i(z) = \sum_{j=0}^{t-1} a_{ij} z^j \bmod q, f_i'(z) = \sum_{j=0}^{t-1} b_{ij} z^j \bmod q$ を生成する．また，$c_{ik} = g^{a_{ik}} h^{b_{ik}} \bmod p$ $(0 \leq k \leq t-1)$ を公開する．さらに，分散情報 $s_{ij} = f_i(j), s_{ij}' = f_i'(j) \bmod q$ $(1 \leq j \leq n)$ を計算し，$s_{ij}, s_{ij}'$ をプレイヤー $P_j$ に送る．

   *Step 2.* 各プレイヤー $P_j$ は，受け取った $s_{ij}, s_{ij}'$ $(1 \leq i \leq n)$ の正当性を確認するため，$g^{s_{ij}} h^{s_{ij}'} = \prod_{k=0}^{t-1} (c_{ik})^{j^k} \bmod p$ $(1 \leq i \leq n)$ が成り立つか検証する．

   *Step 3.* ここで，秘密情報は $x = \sum_{P_i} a_{i0} \bmod q$ であらわされ，各プレイヤー $P_i$ のそれに対する分散情報は $x_i = \sum_{P_j} s_{ji} \bmod q, x_i' = \sum_{P_j} s_{ji}' \bmod q$ となる．

2. 公開鍵生成: 各プレイヤー $P_i$ は，$A_{ik} = g^{a_{ik}} \bmod p$ $(0 \leq k \leq t-1)$ を公開する．プレイヤー $P_j$ は，$P_i \in \mathcal{A}$ に対して，$g^{s_{ij}} = \prod_{k=0}^{t-1} (A_{ik})^{j^k} \bmod p$ が成り立つか検証する．プレイヤー $P_i$ は，$v_i = A_{i0} = g^{a_{i0}} \bmod p$ とし，$y = \prod_{P_i} v_i \bmod p$ を計算し，秘密情報 $x$ に対応する公開鍵とする．

### 2.7.3 暗号化検証

暗号化検証とは，あるデータが平文 $m$ の暗号文であることを証明する手法である．ElGamal 暗号 $E(m) = (e_1, e_2)$ においては，暗号化に用いた乱数の知識証明によって正しい暗号文であることを証明できる．このとき，暗号文の作成者は $PK\{(\alpha) : e_1 = g^\alpha \wedge e_2/m = y^\alpha\}$ によってそれを示すことができる．

### 2.7.4 閾値分散復号

閾値分散復号は，$n$ 人のプレイヤー間に復号関数 $D$ の秘密鍵を分散し，互いに秘密鍵を明かすことなく $t$ $(\leq n)$ 人のプレイヤーが集まることによってのみ，暗号文 $E(m)$ の復号をおこなう手法である．ElGamal 暗号 $E(m) = (e_1, e_2)$ の閾値分散復号の手順は以下のとおり：

初期設定. 各プレイヤーは，分散情報生成プロトコル [25] によって復号関数 $D$ の秘密鍵 $x$ の分散情報を保持するものとする．各プレイヤー $P_i$ の保持する分散情報 $x_i$ に対応する公開鍵を $y_i := g^{x_i} \bmod p$ とする．

*Step 1.* 各プレイヤー $P_i$ は，$(e_1, e_2) \in E_m$ に対し，$s_i = {e_1}^{x_i} \bmod p$ を公開し，$SPK\{(\alpha) : y_i = g^\alpha \wedge s_i = {e_1}^\alpha\}(*)$ によって，$s_i$ が正しく生成されていることを示す.

*Step 2.* プレイヤーたちは，ラグランジェ係数 $\lambda_i = \prod_{j \neq i} x_j/(x_j - x_i)$ を用いて，${e_1}^x = \prod_{i=1}^t {s_i}^{\lambda_i}$ を計算する．

*Step 3.* ${e_1}^x/e_2 = m$ を計算することで，平文 $m$ を得る．

### 2.7.5 分散平文等価テスト

分散平文等価テスト (Plaintext Equality Test)[28] は，2 つの暗号文 $E(m) = (e_1, e_2), E(m') = (e_1', e_2')$ の入力に対し，$t$ $(\leq n)$ 人のプレイヤーが集まることによって，暗号文を復号することなく $m = m'$ か否かのみを判定する手法である．ElGamal 暗号 $E(m) = (e_1, e_2)$ における分散平文等価テストの手順は以下のとおり：

初期設定. 各プレイヤーは，分散情報生成プロトコル [25] を用いて分散された，閾値分散復号に必要な分散情報 $x_i$ を保持し，$y_i := g^{x_i} \bmod p$ を公開しておくものとする．

*Step 1.* それぞれのプレイヤー $P_i$ は，$(t_1, t_2) = (e_1 {e'_1}^{-1}, e_2 {e'_2}^{-1})$ に対して，乱数 $r_i$ を用いて $(z_1^{(i)}, z_2^{(i)}) := ({t_1}^{r_i}, {t_2}^{r_i})$ を生成し，$SPK\{(\alpha) : {z_1}^{(i)} = {t_1}^{\alpha} \wedge {z_2}^{(i)} = {t_2}^{\alpha}\}$ によって，その正当性を示す．

*Step 2.* プレイヤーたちは $z_1 = \prod_{i=1}^{t} {z_1}^{(i)}, z_2 = \prod_{i=1}^{t} {z_2}^{(i)}$ を閾値分散復号し，平文 $\tilde{m}$ を手に入れる．このとき，$\tilde{m} = 1$ ならば，$m = m'$ とし，そうでないならば，$m \neq m'$ とする．

### 2.7.6 ミックス・アンド・マッチ

ミックス・アンド・マッチ [28] は，$t$ $(\leq n)$ 人のプレイヤーによって，0 または 1 の暗号文 $E(m_1), E(m_2)$ に対して，それらの平文の情報を知ることなく，平文同士のビット演算 $G$ の結果の暗号文 $E(G(m_1, m_2))$ を，テーブル $T_{(G)}$ を用いて出力する手法である．このとき，$E(m_1)$ と $E(m_2)$ に演算を施した結果 (乗算など) を入力 $x$ としたミックス・アンド・マッチの出力を $T_{(G)}(x)$ とあらわすことにする．また，$u \in \{0,1\}^*$ に対して 0 に対応する暗号文に $E(1)$，1 に対応する暗号文に $E(u)$ を用いることとする：

*Step 1.* 入力値 $x$ の取り得る値を左列，ビット演算結果 $G(m_1, m_2)$ の暗号文を右列に対応させたテーブル $T_{(G)}$ を作成する．また，暗号化検証によってテーブルを正しく作成したことを示す．$T_{(and)}$ においては，平文同士の乗算結果の暗号文 $E(m_1)E(m_2)$ を左列とし，$E(m_1 \wedge m_2)$ を右列とするテーブル 2.1 を作成する．

*Step 2.* $n$ 人のプレイヤーは，全ての行をシャッフルし，各要素に $E(1)$ を掛けることで再暗号化をおこなう．ここで，$i$ 番目の行を $(left^{(i)}, right^{(i)})$ とする．

*Step 3.* 次に，$t$ 人のプレイヤーは平文等価テストを用いて，入力値と平文が等価である列 $i$ を検索し，その正当性を示す．$AND$ 演算の場合は入力値 $E(m_1)E(m_2)$ との平文等価テストによって，$left^{(i)} \in E_{m_1 m_2}$ をみたす列 $i$ を検索する．

*Step 4.* $right^{(i)}$ を $T_{(G)}(x)$ として出力する.

表 2.1: $T_{(and)}$

| $left^{(i)}$ | $right^{(i)}$ |
|---|---|
| $E(1)$ | $E(1)$ |
| $E(u)$ | $E(1)$ |
| $E(u^2)$ | $E(u)$ |

### 2.7.7 金持ちの財産比べプロトコル

金持ちの財産比べプロトコル [53] は, 2進数であらわされた $k$ ビットの値 $\boldsymbol{b}_1, \boldsymbol{b}_2$ ($\boldsymbol{b}_i = (b_i^{(k-1)}, \ldots, b_i^{(0)})$ の大小関係の比較をおこなう手法である.

*Step 1.* $a_k = 1$ とし, $k-1 \geq j \geq 1$ に対して,

$$s_j := \begin{cases} 1 & b_1^{(j)} = b_2^{(j)} \text{のとき} \\ 0 & \text{それ以外} \end{cases}$$

$$a_j := a_{j+1} \wedge s_j$$

とする.

*Step 2.* $k-1 \geq j \geq 0$ に対し,

$$t_j := \begin{cases} 1 & b_1^{(j)} > b_2^{(j)} \text{のとき} \\ 0 & \text{それ以外} \end{cases}$$

$$u_j := a_{j+1} \wedge t_j$$

とする.

*Step 3.*

$$v := u_0 \vee \cdots \vee u_{k-1}$$

を求める.

*Step 4.* $v = 1$ のとき, $\boldsymbol{b}_1 < \boldsymbol{b}_2$ とし, $v = 0$ のとき, $\boldsymbol{b}_1 \geq \boldsymbol{b}_2$ とする.

### 2.7.8 ビット・スライスプロトコル

ビットスライスプロトコル [31] は，$m$ 個の値 $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_m$ ($\boldsymbol{b}_i = (b_i^{(k-1)}, \ldots, b_i^{(0)})$) から最大値 $\boldsymbol{b}_{max}$ のみを出力する手法である．ここでは，代理人入札の価格更新への適用を考え，$m = 2$ として $\boldsymbol{b}_1, \boldsymbol{b}_2$ から低い方の値 $\boldsymbol{b}_{low} = (b^{(k-1)}, \ldots, b^{(0)})$ を出力する手法を述べる．

*Step 1.* $\boldsymbol{w} = (0, 0)$ とし，$j = k - 1$ から 0 まで以下を繰り返す;

*Step 1-1.* $\boldsymbol{w} = (w_1, w_2)$ に対して，

$$\boldsymbol{s}_j := (w_1 \vee b_1^{(j)}, w_2 \vee b_2^{(j)})$$

とする．

*Step 1-2.* $\boldsymbol{s}_j = (s_1, s_2)$ に対して，

$$b^{(j)} := s_1 \wedge s_2$$

とする．

*Step 1-3.* $b^{(j)}$ の結果を受けて，$\boldsymbol{w}$ に

$$\boldsymbol{w} := \begin{cases} \boldsymbol{w} & b^{(j)} = 1 \text{ のとき} \\ \boldsymbol{s}_j & \text{それ以外} \end{cases}$$

を代入し，$j = j - 1$ とし，*Step 1* へ．

*Step 2.* $\boldsymbol{w} = (w_1, w_2)$ に対して，$w_i = 0$ をみたす $\boldsymbol{b}_i$ を $\boldsymbol{b}_{low} = \boldsymbol{b}_i = (b^{(k-1)}, \ldots, b^{(0)})$ として出力する．

# 第 3章

# 既存方式

本章では，匿名証明書方式と電子オークションの既存方式を説明する．匿名証明書方式では，既存の方式の中でも最も効率のよいといわれている Camenisch らによる手法 [11] を紹介し，問題点を挙げる．また，代理入札方式は電子オークションの一種であり，ここでは電子オークションの代表的な手法としてイングリッシュ・オークションを挙げ，第二価格入札を用いた代理入札方式の実現手法を述べる．

なお，電子保証人方式については既存の提案方式が存在しないため，4 章にて既存技術を用いて電子保証人方式に必要な性質が実現できるか考察する．

## 3.1 既存の匿名証明書方式

匿名証明書方式 [15, 17, 18, 7, 32, 11] では，ユーザが異なる仮名 (Pseudonym) を取引機関へ登録し，各機関はその仮名によってユーザの情報を管理・識別するため，それぞれの機関のもつ情報の結合による個人情報の漏洩・拡大を防ぐことができる．また，各機関は登録されたユーザの仮名に対する証明書 (Credential) を発行し，対応するユーザは証明書の保持証明をおこなうことで，検証者にその機関の登録者であるという事実のみを示すことができる．Lysyanskaya, Rivest, Sahai, Wolf [32] は，このような匿名証明書方式を一方向性関数の性質を用いて実現している．しかし，彼らの手法では証明書の保持証明を 1 度しか行うことができず，複数回の証明はユーザに関する情報を検証者に漏らしてしまう．よって，ユーザは登録証明のつど，仮名に対する新たな証明書を発行してもらう必要があった．そ

こで，Camenisch と Lysyanskaya [11] は登録証明書を強 RSA 問題に基づく署名とし，その保持証明によって機関への登録を証明するという手法を提案した．これによって，ユーザは証明書を保持しているという事実以外の情報を何ら漏らすことなく，複数回の登録証明をおこなうことができる．

彼らの手法は以下のエンティティで構成される:

機関 $O_i$: 秘密鍵，公開鍵のペアをもち，それらを用いてユーザに登録証明書を発行するエンティティ．

ユーザ $U$: 機関に仮名を登録し，証明書の発行をうけるエンティティ．

検証者 $V$: ユーザの属性を検証するエンティティ．

既存方式の流れを以下に紹介する．

1. 初期設定:

   各機関 $O_i$ はそれぞれ秘密鍵，公開鍵の組を生成する．$O_i$ によって生成される公開鍵を RSA の法 $n_{O_i}$ と $d_{O_i}, e_{O_i}, f_{O_i}, g_{O_i}, h_{O_i} \in {\mathbb{Z}^*_{n_{O_i}}}^2$ とし，秘密鍵を $n_{O_i}$ を生成する素数とする．また，ユーザ $U$ はシステム内で用いる秘密鍵 $x_U$ を生成する．

2. ユーザの機関への登録:

   ユーザ $U$ は仮名生成プロトコルを用いて，機関 $O_i$ に仮名 $P_{(U,O_i)}$ を登録する．これは $U$ の秘密鍵 $x_U$ を用いて $P_{(U,O_i)} := {g_{O_i}}^{x_U} {h_{O_i}}^{s_{(U,O_i)}} \bmod n_{O_i}$ で生成される．但し，$s_{(U,O_i)}$ は $U$ の秘密情報である．また，$O_i$ は $P_{(U,O_i)}$ に対して $C_{(U,O_i)} \equiv (P_{(U,O_i)} f_{O_i})^{1/E_{(U,O_i)}} \pmod{n_{O_i}}$ をみたす $(E_{(U,O_i)}, C_{(U,O_i)})$ の組を匿名証明書として発行する．

3. 検証者への登録証明:

   ユーザは登録証明プロトコルによって，

   $$ {C_{(U,O_i)}}^{E_{(U,O_i)}} \equiv {g_{O_i}}^{x_U} {h_{O_i}}^{s_{(U,O_i)}} f_{O_i} \pmod{n_{O_i}} $$

   をみたす $C_{(U,O_i)}, E_{(U,O_i)}, x_U, s_{(U,O_i)}$ の知識証明によって検証者に $O_i$ の匿名証明書の保持を示すことができる．

このプロトコルは，$\alpha^{\beta} \equiv {g_{O_i}}^{\gamma} {h_{O_i}}^{\delta} f_{O_i} \pmod{n_{O_i}}$ をみたす $\alpha, \beta$ の組を生成できるのは，合成数 $n_{O_i}$ の素因数を知るエンティティ，即ち機関 $O_i$ であること，またそれらが ${g_{O_i}}^{\gamma} {h_{O_i}}^{\delta} f_{O_i}$ に対して生成されていることから，$\gamma, \delta$ の知識証明は $O_i$ に登録されている仮名に対応するユーザであることを意味する．このプロトコルは，$(\alpha, \beta, \gamma, \delta) = (C_{(U,O_i)}, E_{(U,O_i)}, x_U, s_{(U,O_i)})$ を見せることなく，その事実を検証者に証明することができるため，複数回の登録証明でさえ，それらが同じユーザによっておこなわれているという情報を検証者に与えない．

4. 検証機関への匿名証明:

ユーザは別機関への匿名証明プロトコルによって，異なる仮名 $P_{(U,O_j)}$ を登録してある別機関 $O_j$ に，

$$C_{(U,O_i)}{}^{E_{(U,O_i)}} \equiv {g_{O_i}}^{x_U} {h_{O_i}}^{s_{(U,O_i)}} f_{O_i} \pmod{n_{O_i}}$$

$$P_{(U,O_j)} \equiv {g_{O_j}}^{x_U} {h_{O_j}}^{s_{(U,O_j)}} \pmod{n_{O_j}}$$

を同時にみたす $C_{(U,O_i)}, E_{(U,O_i)}, x_U, s_{(U,O_i)}, s_{(U,O_j)}$ の知識証明によって，$O_i$ の登録証明をおこなうとともに，$P_{(U,O_j)}$ に対応するユーザであることを証明することができる．これは，$O_i$ によって発行された登録証明書に対応する仮名と $P_{(U,O_j)}$ が同じ秘密鍵を用いて生成されていることを意味する．

各プロトコルの構築は以下で与えられる．

### 初期設定

システムパラメータを以下のように設定する．RSA の法のサイズを $\ell_n$ ビット，セキュリティパラメータを $\epsilon > 1$ する．また，$\ell_\Gamma = 2\ell_n$, $\ell_\Delta = \epsilon\ell_\Gamma$, $2^{\ell_\Lambda} > 2(2^{2\ell_\Gamma} + 2^{\ell_\Gamma} + 2^{\ell_\Delta})$, $2(2^{\ell_\Sigma}(2^{2\ell_\Gamma} + 2^{\ell_\Delta}) + 2^{\ell_\Delta}) < 2^{\ell_\Lambda}$ とし，各区間を $\Gamma = ]-2^{\ell_\Gamma}, 2^{\ell_\Gamma}[$, $\Delta = ]-2^{\ell_\Delta}, 2^{\ell_\Delta}[$, $\Lambda = ]2^{\ell_\Lambda}, 2^{\ell_\Lambda+\ell_\Sigma}[$ と設定する．

### 鍵生成プロトコル

*Step 1.* 各機関 $O_i$ は，$p_{O_i} := 2p'_{O_i} + 1$ かつ $q_{O_i} := 2q'_{O_i} + 1$ が素数となるような $\ell_n/2$ ビットの素数 $p'_{O_i}, q'_{O_i}$ を選択し，$n_{O_i} := p_{O_i} q_{O_i}$ を計算する．

また，$d_{O_i}, e_{O_i}, f_{O_i}, g_{O_i}, h_{O_i} \in_R {\mathbb{Z}^*_{n_{O_i}}}^2$ を選択し，$(p_{O_i}, q_{O_i})$ を秘密鍵，$(n_{O_i}, d_{O_i}, e_{O_i}, f_{O_i}, g_{O_i}, h_{O_i})$ をそれに対応する公開鍵とする．

*Step 2.* ユーザ $U$ は，システム内で用いる秘密鍵 $x_U \in_R \Gamma$ を生成し，保管する．

### 仮名生成プロトコル

機関 $O_i$ に仮名を登録するため，ユーザ $U$ は以下のプロトコルを実行する:

*Step 1.* $U$ は $r_1 \in_R \Delta, r_2, r_3 \in_R \{0,1\}^{2\ell_n}$ を選び，$c_1 := {d_{O_i}}^{r_1} {e_{O_i}}^{r_2}, c_2 := {d_{O_i}}^{x_U} {e_{O_i}}^{r_3}$ を計算し，$O_i$ に送る．

$$PK^2\{(\alpha, \beta, \gamma, \delta) \ : \ c_1 = {d_{O_i}}^{\alpha} {e_{O_i}}^{\beta} \wedge c_2 = {d_{O_i}}^{\gamma} {e_{O_i}}^{\delta}\}$$

によって，$c_1, c_2$ が正しく生成されていることを証明する．

*Step 2.* $O_i$ は $r \in_R \Delta$ を選択し，$U$ に送信する．

*Step 3.* $U$ は，$s_{(U,O_i)} := (r_1 + r \pmod{2^{\ell_\Delta + 1} + 1}) - 2^{\ell_\Delta} + 1$ と $\tilde{s} = \lfloor r_1 + r/(2^{\ell_\Delta + 1} - 1) \rfloor$ を計算し，$P_{(U,O_i)} := {g_{O_i}}^{x_U} {h_{O_i}}^{s_{(U,O_i)}}$ を仮名とする．また $r_4 \in_R \{0,1\}^{\ell_n}$ に対し $c_3 := {d_{O_i}}^{\tilde{s}} {e_{O_i}}^{r_4}$ と $P_{(U,O_i)}$ を $O_i$ に送信し，それが正しく生成されたことを以下の $PK^2$ によって証明する:

$$\begin{aligned} PK^2\{(\alpha, \beta, \gamma, \delta, \varepsilon, \zeta, \vartheta, \xi) \quad : \quad & c_1 = {d_{O_i}}^{\alpha} {e_{O_i}}^{\beta} \\ \wedge \quad & c_2 = {d_{O_i}}^{\gamma} {e_{O_i}}^{\delta} \\ \wedge \quad & c_3 = {d_{O_i}}^{\varepsilon} {e_{O_i}}^{\zeta} \\ \wedge \quad & P_{(U,O_i)} = {g_{O_i}}^{\gamma} {h_{O_i}}^{\vartheta} \\ \wedge \quad & (c_1 d^{r - 2^{\ell_\Delta + 1}}) / ({c_3}^{2^{\ell_\Delta + 1}} + 1) = {d_{O_i}}^{\vartheta} {e_{O_i}}^{\xi} \\ \wedge \quad & \gamma \in \Gamma \wedge \vartheta \in \Delta\}. \end{aligned}$$

*Step 5.* $O_i$ は仮名リストに $P_{(U,O_i)}$ を保管する．

*Step 6.* $U$ は $P_{(U,O_i)}$ の生成に用いた秘密情報 $s_{(U,O_i)}$ と $P_{(U,O_i)}$ を $O_i$ の登録情報として秘密に保管する．

**登録証明書発行プロトコル**

機関 $O_i$ は，ユーザ $U$ の仮名 $O_{(U,O_i)}$ に対して，登録証明書を発行するため，以下のプロトコルを実行する:

*Step 1.* $U$ は $PK^2\{(\alpha,\beta) : P_{(U,O_i)} = {g_{O_i}}^{\alpha}{h_{O_i}}^{\beta}\}$ によって $O_i$ のデータベースに登録されている $P_{(U,O_i)}$ に対応するユーザであることを証明する．

*Step 2.* $O_i$ は素数 $E_{(U,O_i)} \in_R \Lambda$ を選択し，$C_{(U,O_i)} := (P_{(U,O_i)} f_{O_i})^{1/E_{(U,O_i)}} \pmod{n_{O_i}}$ を計算する．また $U$ に $O_i$ の登録証明書として $(E_{(U,O_i)}, C_{(U,O_i)})$ を送る．$O_i$ は $(E_{(U,O_i)}, C_{(U,O_i)})$ をそれに対応する仮名 $P_{(U,O_i)}$ とともに保管する．

*Step 3.* $U$ は ${C_{(U,O_i)}}^{E_{(U,O_i)}} \equiv P_{(U,O_i)} f_{O_i} \pmod{n_{O_i}}$ かつ $E_{(U,O_i)} \in \Lambda$ であることを検証し，正しければ $\mathcal{C}_{(U,O_i)} = (E_{(U,O_i)}, C_{(U,O_i)})$ を機関 $O_i$ の登録証明書として保管する．

**登録証明プロトコル**

ユーザ $U$ は機関 $O_i$ によって発行された登録証明書を保持し，さらにそれに対応するユーザであることを証明する．これにより，$O_i$ への登録者であることを検証者 $V$ に示すことができる．

*Step 1.* $U$ は $r_1, r_2 \in_R \{0,1\}^{2\ell_n}$ を選択し $c_1 := C_{(U,O_i)} {e_{O_i}}^{r_1}, c_2 := {e_{O_i}}^{r_1} {d_{O_i}}^{r_2}$ を計算し，$V$ に $c_1, c_2$ を送る．

*Step 2.* $U$ は以下の $PK^2$ によって $O_i$ への登録証明をおこなう:

$$\begin{aligned} PK^2\{(\alpha,\beta,\gamma,\delta,\varepsilon,\zeta,\xi) \quad : \quad & f_{O_i} = {c_1}^{\alpha} / {g_{O_i}}^{\beta} {h_{O_i}}^{\gamma} {e_{O_i}}^{\delta} \\ \wedge \quad & c_2 = {e_{O_i}}^{\varepsilon} {d_{O_i}}^{\zeta} \\ \wedge \quad & 1 = {c_2}^{\alpha} / {e_{O_i}}^{\delta} {d_{O_i}}^{\xi} \\ \wedge \quad & \alpha \in \Lambda \wedge \beta \in \Gamma \wedge \gamma \in \Delta\}. \end{aligned}$$

**検証機関への登録証明プロトコル**

ユーザ $U$ は，検証機関 $O_j$ に対して機関 $O_i$ への登録を証明するとともに，$O_j$ に登録してある仮名 $P_{(U,O_j)}$ に対応するユーザであることを示す．これにより，$P_{(U,O_j)}$

が $O_j$ への登録者であることを $O_j$ に示すことができる.

*Step 1.* ユーザ $U$ は $r_1, r_2 \in_R \{0,1\}^{2\ell_n}$ を選択し，$c_1 := C_{(U,O_i)}{e_{O_i}}^{r_1}$ と $c_2 := {e_{O_i}}^{r_1}{d_{O_i}}^{r_2}$ を計算し，$c_1, c_2$ を検証機関 $O_j$ に送る.

*Step 2.* $U$ は以下の $PK^2$ によって $P_{(U,O_j)}$ による $O_i$ への登録証明をおこなう．これにより，登録証明書に対応する仮名と $P_{(U,O_j)}$ が同じ秘密鍵を用いて生成されていることが証明できる:

$$\begin{aligned} PK^2\{(\alpha,\beta,\gamma,\delta,\varepsilon,\zeta,\xi,\eta) \quad : \quad & f_{O_i} = {c_1}^{\alpha}/{g_{O_i}}^{\beta}{h_{O_i}}^{\gamma}{e_{O_i}}^{\delta} \\ \wedge \quad & c_2 = {e_{O_i}}^{\varepsilon}{d_{O_i}}^{\zeta} \\ \wedge \quad & 1 = {c_2}^{\alpha}/{e_{O_i}}^{\delta}{d_{O_i}}^{\xi} \\ \wedge \quad & P_{(U,O_j)} = {g_{O_j}}^{\beta}{h_{O_j}}^{\eta} \\ \wedge \quad & \alpha \in \Lambda \wedge \beta \in \Gamma \wedge \gamma \in \Delta\}. \end{aligned}$$

この方式における登録証明書は，各機関の設定した群 ${\mathbb{Z}^*_{n_{O_i}}}^2$ 上で生成された RSA 署名であり，機関の公開鍵を用いずに保持証明をおこなうことはできない．よって，機関の特性のみを検証者に示したい場合には，機関をグループ化し，そのグループ管理者によって別の属性証明書を生成してもらう必要がある．また，$m$ 個の登録証明をおこなう場合にも，独立に $m$ 回の $PK^2$ プロトコルをおこなわなければならない.

## 3.2 電子オークション

オークションには，価格吊り上げ型のイングリッシュ・オークション [42, 43] と全ての入札値を一度に比較するシールド・ビッド・オークション [30, 49, 28, 31], 第二価格入札 [26, 37, 29, 5] らがある.

### 3.2.1 イングリッシュ・オークション

イングリッシュ・オークションでは，条件を満たす入札値が現在の落札価格 (公開) となるため，入札値の秘匿性は要求されない．しかし，参加者のプライバシ確保の観点から，入札値と入札者の関係，及び落札者以外の入札者を秘匿する必要

がある．また，入札，及び価格更新はリアルタイムでおこなわれるため，1 回の入札・価格更新にかかる通信量，計算量コストの削減が非常に重要になる．ここで，面・宮地によるイングリッシュオークション [42, 43] の概要を述べる．このシステムはオークション参加者の登録・削除をおこなう登録管理者 $RM$，オークション管理者 $AM_1, AM_2$，参加者 $B_i$ で構成され，以下のプロトコルからなるものである:

1. 初期設定: オークション管理者は，暗号化関数とともに公開鍵を生成し，それらを公開する．また，復号関数の秘密鍵は 2 人のオークション管理者で分散する．

2. オークション準備: 登録管理者とオークション管理者はオークション毎に，各参加者に必要な情報を設定し，それらを公開する．また，オークション管理者はそれぞれのオークションに関する，[商品，出品者情報，オークション開始価格，現在の落札価格，オークション終了までの時間] などを公開する．

3. 参加者登録: 参加者は，オークションに参加するため，登録鍵を登録し，オークション参加に必要となる情報を取得する．

4. 入札: 参加者 $B_i$ は，オークション管理者の公開鍵暗号で暗号化した入札値を送る．

5. 価格更新 オークション管理者は暗号化された入札値を復号し，それを現在の落札価格とする．

6. 落札者決定: オークション管理者は落札者に関する情報を登録管理者に送り，登録管理者はその情報から落札者を決定する．

このようなイングリッシュ・オークションの特徴は複数入札が可能なことである．したがって，入札・価格更新の効率性が重要である．これは登録管理者を設けることで，入札を暗号化処理のみでおこなうことができるが，入札値と入札者の関係を漏洩しない方式である．

### 3.2.2 シールド・ビッド・オークション，第二価格入札

シールド・ビッド・オークション，第二価格入札は，各参加者が自分の入札値を秘密にしたまま，1 度だけ入札をおこなうオークションであり，それぞれ全ての入札値における最高入札値，2 番に高い入札値が落札額となるオークションである．よって，公平なオークションをおこなうためには，各入札値は秘匿されなければならず，参加者のプライバシ確保の観点からは，開札後の落札額以外の秘匿性，及び落札者以外の匿名性が必要である．

一方，価格更新に第二価格入札を適用することで，代理入札を実現することができる．この場合，2 つの入札値に対して第二価格入札をおこなうので，高い方の入札値 (最高額) を秘匿したまま，低い方の入札値 (2 番目に高い値) のみを得ることができる．

ここで，阿部・鈴木による $(M+1)$-st オークション [5] を，代理入札に適用することを考える．このシステムも登録管理者 $RM$，オークション管理者 $AM_1, AM_2$，参加者 $B_i$ で構成される:

1. 初期設定: オークション管理者 $AM_1$ は公開鍵暗号 $E$ を選択し，$u \in \{0,1\}^*$，及び離散値価格リスト $List = \{price_\ell, \ldots, price_1\}$ を公開する．

2. 入札: 参加者 $B_i$ は落札希望額 $price_{k_i} \in List$ に対し，$\Delta E(\boldsymbol{b}_i) = (\Delta e_i^{(\ell)}, \ldots \Delta e_i^{(1)})$ を計算する．但し，各ビット $\Delta e_i^{(j)}$ を

$$\Delta e_i^{(j)} = \begin{cases} E(u) & j = k_i \text{ のとき} \\ E(1) & j \neq k_i \text{ のとき} \end{cases}$$

とする．さらに，入札値 $\Delta E(\boldsymbol{b}_i)$ の正当性を証明するために

$$\Delta e_i^{(\ell)} \cdots \Delta e_i^{(1)} \in E_u \ (1 \leq j \leq \ell),$$

$$\Delta e_i^{(j)} \in \{E_1 \cup E_u\} \ (1 \leq j \leq \ell)$$

の知識証明を生成し，$\Delta E(\boldsymbol{b}_i)$ と共に公開する．

3. 価格更新: 2 人のオークション管理者 $AM_1, AM_2$ は，参加者 $B_{new}$ による新たな入札値 $\Delta E(\boldsymbol{b}_{new}) = (\Delta e_i^{(\ell)}, \ldots, \Delta e_i^{(1)})$ と，現在の落札者 $B_{high}$ の入札値

$E(\boldsymbol{b}_{high}) = (e^{(\ell)}_{high}, \ldots, e^{(1)}_{high})$, ($E(\boldsymbol{b}_{high})$ は前ステージ以前に求められている) からそれらのうち，低い入札値

$$price_{low} = \begin{cases} price_{k_i} & price_{k_i} < price_{k_h} \text{のとき} \\ price_{k_h} & price_{k_i} \geq price_{k_h} \text{のとき} \end{cases}$$

を求める．但し，$B_{high}$ の落札希望額を $price_{k_h}$ とする．

*Step 1.* $\Delta E(\boldsymbol{b}_{new})$ に対し，$AM_1, AM_2$ はそれぞれ

$$\begin{aligned} e^{(\ell)}_{new} &:= \Delta e^{(\ell)}_{new}, \\ e^{(\ell-1)}_{new} &:= \Delta e^{(\ell-1)}_{new} \cdot e^{(\ell)}_{new}, \\ &\vdots \qquad\quad \vdots \\ e^{(1)}_{new} &:= \Delta e^{(1)}_{new} \cdot e^{(2)}_{new}. \end{aligned}$$

を計算し，$E(\boldsymbol{b}_{new}) = (e^{(\ell)}_{new}, \ldots, e^{(1)}_{new})$ を生成する．

*Step 2.* $E(\boldsymbol{b}_{new})$, $E(\boldsymbol{b}_{high})$ に対し，

$$c_j := e^{(j)}_{new} \cdot e^{(j)}_{high} \ (1 \leq j \leq \ell)$$

を計算する．

*Step 3.* 各 $c_j$ に対し，平文等価テストを用いたバイナリ・サーチによって $c_j \in E_{u^2}$ かつ $c_{j+1} \notin E_{u^2}$ をみたす $price_{low} = j$ を検索し，出力する．

4. 落札者決定 オークション管理者 $AM_1$ と $AM_2$ は，オークション終了時の $(e^{(j+1)}_{new}, e^{(j+1)}_{high})$ をそれぞれ閾値分散復号し，その復号結果を $(w_1, w_2)$ としたとき，

$$B_{high} = \begin{cases} B_{new} & (w_1, w_2) = (u, 1) \text{ のとき} \\ B_{high} & \text{それ以外} \end{cases}$$

とし，$B_{high}$ を落札者として $price_{low}$ とともに公開する．

このような第二価格入札は，元来，一斉入札後の処理を対象としているため，落札者決定に必要な計算量が大きく，リアルタイムでの処理が必要な代理入札には適さない．よって，オークションのみたすべき性質を損なわず，価格更新を効率よくおこなうことのできる代理入札システムの提案が必要である．

# 第 4章

# 電子保証人方式

## 4.1 はじめに

電子商取引への保証の形態として，取引内容に対する保証と取引ユーザとの取引そのものに対する保証が挙げられるが，前者はユーザの文書 (取引内容) に対する保証であり，後者はユーザ自身の保証である．ユーザ自身に対する保証とはユーザの生成する全ての取引内容への保証を意味する．一般に，取引内容に対する保証者の役割はユーザの役割を包含し，実社会では誰が保証者であるかが検証できれば十分である状況が多く考えられる．このことから，電子保証において，ユーザと保証者の立場の相違が明確であり，保証者のみの検証は可能であるが，ユーザのみの検証は不可能であるという性質を実現できる必要がある．

また，ディジタル署名の技術を用いることで文書の正当性を証明できることから，このような電子保証はユーザのディジタル署名に対する保証と，ユーザの全てのディジタル署名に対する保証の 2 形態に分類することができる．以後，これらを取引に対する電子保証，ユーザに対する電子保証とよぶことにする．

ここで，ユーザを $U$，保証者を $G$ とした電子保証人方式の満たすべき性質を以下のように定義する．

電子保証人付署名の正当性: $U$ の $G$ による電子保証付署名はユーザ $U$ と保証者 $G$ の同意の下でのみ生成される．

署名の偽造不可能性: $U$ 以外のだれも，$U$ の署名を生成することはできない．

保証書の偽造不可能性: $G$ 以外のだれも，$G$ による電子保証を与えることはできない．

一般検証可能性: だれでも，$G$ によって電子保証を与えられた $U$ の署名であることを検証できる．

保証単独検証可能性: だれでも $G$ によって電子保証を与えられた署名であることを検証できる．

署名単独検証不可能性: ディジタル署名の正当性のみを検証することはできない．

このような性質をもちあわせる手法を電子保証人方式と定義し，安全でかつ効率的な手法を提案する．

## 4.2 既存技術を用いた電子保証人方式の実現

本節では，電子保証人方式の実現に既存技術を用いることを考える．

複数のエンティティが同一メッセージに順に署名を施す手法である多重署名方式 [33, 35, 51, 39] では，検証者が全ての署名者の公開鍵を用いることによって，それらエンティティの同意の下に生成されたものであることを確認することができる．しかしながら，多重署名方式における全ての署名者は対等であるため，電子保証人方式に必要となる保証単独検証可能性，署名単独検証不可能性が実現できない．

また，公開鍵の正当性を信頼できるセンターに保証してもらう際に用いられる公開鍵証明書方式について考察してみる．この場合，保証者による保証書とはユーザの公開鍵に対するディジタル署名であり，検証者はその署名の対象となる公開鍵を用いてユーザの署名の正当性を確認する．この方式を応用した場合，保証人による保証情報はユーザの公開鍵に対するディジタル署名となるが，このような保証情報がなくとも，ユーザのディジタル署名の正当性を確認することができる．したがって，この方式は署名単独検証不可能性をもたない．

## 4.3 取引に対する電子保証人方式

本節では，取引に対する電子保証人方式を実現し，その安全性を考察する．この場合，保証者の生成する情報である保証書は固定された取引内容 (メッセージ) とユーザの組に対して作成され，ユーザの生成する情報は保証書を用いて作成されるべきである．即ち保証者の公開鍵を用いてのみ署名の正当性の検証をおこなうことができるようにすることで，保証の単独検証を可能にする．

### 4.3.1 プロトコル

取引に対する電子保証人方式を実現するためのプロトコルを以下で与える:

1. 初期設定: $q \mid p-1$ をみたす十分大きな素数 $p, q$ と $q$ を位数とする $g \in \mathbb{Z}_p^*$ を用意する．また，共通鍵暗号化関数，および共通鍵復号関数 $\mathcal{C}_K$，${\mathcal{C}_K}^{-1}$ と，ハッシュ関数 $\mathcal{H} : \{0,1\}^* \to \{0,1\}^{|n|}$ を設定する．

2. 鍵生成: 鍵生成アルゴリズム

$$KG_U(1^n) \ni (x_U, y_U), KG_G(1^n) \ni (x_G, y_G)$$

はセキュリティパラメータ $n$ に対し，$1^n$ の入力によって，鍵の組を出力する多項式時間アルゴリズムである．出力される $(x_U, y_U)$ と $(x_G, y_G)$ は，それぞれ $U$ と $G$ の秘密鍵と公開鍵の組である．ここで，$KG(1^n) \ni (x, y)$ は $(x, y)$ がアルゴリズム $KG$ への入力 $1^n$ によって出力されることを意味する．

– ユーザ $U$，保証者 $G$ ともに，秘密鍵 $x_U \in_R \mathbb{Z}_q^*, x_G \in_R \mathbb{Z}_q^*$ を生成し，$y_U := g^{x_U} \bmod p, y_G := g^{x_G} \bmod p$ をそれぞれ公開鍵とする．

3. 電子保証付署名生成: 電子保証付署名生成プロトコル

$$InSig\langle U(x_U), G(x_G)\rangle(m, y_U, y_G) \ni \sigma_{UG}$$

は，$U$ によるプライベート入力 $x_U$，$G$ によるプライベート入力 $x_G$，共通入力 $m, y_U, y_G$ に対してメッセージ $m$ に対する $U$ の $G$ による保証付署名 $\sigma_{UG}$ を出力する．

*Step 1.* ユーザ $U$ は，$r_U \in_R \mathbb{Z}_q^*$ に対して $t_U := g^{r_U} \bmod p$ を計算し，メッセージ $m$ と $t_U$ を保証者 $G$ に送る．

*Step 2.* 保証者 $G$ は，$r_G \in_R \mathbb{Z}_q^*$ を選択し，$t_G := g^{r_G} \bmod p, R := 0^n \,||\, y_U$ を計算し，$t_G$ を鍵として用いた $R$ の暗号文 $c_G := \mathcal{C}_{t_G}(R), e := \mathcal{H}(m, t_U y_U t_G, c_G)$ を求める．また，$s_G := r_G - e x_G \bmod q$ とし $(e, s_G, c_G)$ を $U$ に送る．

*Step 3.* $U$ は $\tilde{t}_G := g^{s_G} {y_G}^e$ に対し，$e = \mathcal{H}(m, t_U y_U \tilde{t}_G, c_G)$ が成り立つかどうか検証し，正しければ $s_U := r_U - e x_U \bmod q$ とし，メッセージ $m$ に対する署名として $\sigma_{UG} = (e, s_U, s_G, c_G)$ を出力する．

4-1. 一般検証: 一般検証アルゴリズム

$$InVer_{(U,G)}(m, \sigma_{UG}, y_U, y_G) \ni \{1, 0\}$$

は，メッセージ $m$，署名 $\sigma_{UG}$，ユーザと検証者の公開鍵 $y_U, y_G$ に対し，$\sigma_{UG} \in InSig\langle U(x_U), G(x_G)\rangle(m, y_U, y_G)$ である場合は圧倒的確率で 1 を，そうでない場合は 0 をそれぞれ出力する．

– 検証者 $V$ は，$\sigma_{UG} = (e, s_U, s_G, c_G)$ に対し，$\tilde{t} := g^{s_U + s_G}(y_U y_G)^e y_U \bmod p$ を計算し，$e = \mathcal{H}(m, \tilde{t}, c_G)$ がなりたつならば，$\sigma_{UG}$ を $U$ による $G$ の保証付署名として受理し，そうでない場合には拒否する．

4-2. 保証書単独検証: 保証書単独検証アルゴリズム

$$InVer_G(m, \sigma_{UG}, y_G) \ni \{1, 0\}$$

は，メッセージ $m$，署名 $\sigma_{UG}$，保証者の公開鍵 $y_G$ に対し，$\sigma_{UG} \in InSig\langle U(x_U), G(x_U)\rangle(m, y_U, y_G)$ である場合には圧倒的確率で 1 を出力し，そうでない場合は 0 をそれぞれ出力する．

– 検証者 $V$ は，$\tilde{t}_G := g^{s_G} {y_G}^e \bmod p$ を鍵として $c_G$ を復号し $\tilde{R} := {\mathcal{C}_{\tilde{r}_G}}^{-1}(c_G)$ の上位 $n$ ビットを $b$ とし，$b \,||\, \beta$ に分割する．このとき，$b = 0^n$ かつ，$e = \mathcal{H}(m, g^{s_U} \beta^{e+1} \tilde{t}_G, c_G)$ ならば，$\sigma_{UG}$ を $G$ の保証付署名として受理し，そうでない場合には拒否する．

### 4.3.2 安全性の考察

ここで，取引に対する電子保証人方式の安全性について考察する．

システムパラメータ，公開鍵が与えられ，ランダム・オラクル $\mathcal{H}$ と理想的共通鍵暗号関数 $\mathcal{C}$，または復号関数 $\mathcal{C}^{-1}$ にアクセスを許された攻撃者 $InAdv$ を仮定する．このとき，ユーザ，及び保証者に対する安全性について以下の定理が与えられる．

**定理 1** (ユーザに対する安全性) 保証人 $G$ と結託することで，適応的に選択したメッセージ $m_i$ $(1 \leq i \leq \ell)$ に対する $U_i$ による署名 $\sigma_{U_iG}(m_i)$ を得ることができる攻撃者 $InAdv$ が，$(m, U) \neq \forall(m_i, U_i)(1 \leq i \leq \ell)$ に対して，

$$InVer_{(U,G)}(m, \sigma_{UG}(m), y_U, y_G) = 1 \vee InVer_G(m, \sigma_{UG}(m), y_G) = 1$$

をみたす $\sigma_{UG}(m)$ を生成できる確率は無視できる．

*Proof.* ランダム・オラクル仮定の下，$InAdv$ がメッセージ $m$ に対する $U$ の署名 $\sigma_{UG}(m)$ を偽造することができたと仮定する．このとき，Forking Lemma[?] より $InAdv$ は $U$ の署名 $(t_U, e, s_U)$ と $(t_U, \tilde{e}, \tilde{s_U})$ を得ることができる．このとき，$t_U y_U t_G \equiv g^{s_U+s_G}(y_U y_G)^e y_U \equiv g^{\tilde{s}_U+s_G}(y_U y_G)\tilde{e} y_U \pmod{p}$ より，これらは $t_U = g^{s_U}(y_U y_G)^e = g^{\tilde{s_U}}(y_U y_G)^{\tilde{e}} \bmod p$ をみたすので，$g^{s_U - \tilde{s_U}} = (g^{x_U+x_G})^{e-\tilde{e}} \bmod p$ より，$x_U = (s_U - \tilde{s_U})/(\tilde{e} - e) - x_G \bmod q$ を得る．これは，$y_U, g \in \mathbb{Z}_p^*$ に対する離散対数問題の解であり，離散対数問題の困難さの仮定に反する．従って，$InAdv$ が正当な署名を生成する確率は無視できる．

**定理 2** (保証者に対する安全性) ユーザ $U$ と結託することで，適応的に選択したメッセージ $m_i$ $(1 \leq i \leq \ell)$ に対する $G_i$ による保証付署名 $\sigma_{UG_i}(m_i)$ を得ることができる攻撃者 $InAdv$ が，$(m, G) \neq \forall(m_i, G_i)$ に対して，

$$InVer_{(U,G)}(m, \sigma_{UG}(m), y_U, y_G) = 1 \vee InVer_G(m, \sigma_{UG}(m), y_G) = 1$$

をみたす $\sigma_{UG}(m)$ を生成できる確率は無視できる．

*Proof.* $InAdv$ の動きを以下の 4 つに分類し，それぞれについて考察する．

*Case 1.* $m \in \{0,1\}^*, t_U, y_U, t_G \in_R \mathbb{Z}_p^*, c \in_R \{0,1\}^*$ を $\mathcal{H}$ に質問し，答えとして $e$ を得た後，$t_G$ を鍵とした $R$ の暗号文を $\mathcal{C}$ に求め，$c' = \mathcal{C}_{t_G}(R)$ を得る．このとき，既に選択した $c$ に対して $c = c'$ となる確率は無視できるほど小さい．

*Case 2.* $m \in \{0,1\}^*, t_U, y_U, t_G \in_R \mathbb{Z}_p^*, c \in_R \{0,1\}^*$ を $\mathcal{H}$ に質問し，答えとして $e$ を得た後，$t_G$ を鍵とした $c$ の復号文を $\mathcal{C}^{-1}$ を求め，$R' = \mathcal{C}_{t_G}^{-1}(c)$ を得る．このとき，$R$ が $n$ ビットの冗長をもつ，即ち $0^n \mid\mid \beta$ という形になる確率は無視できるほど小さい．

*Case 3.* $t_G \in_R \mathbb{Z}_q^*, c \in \{0,1\}^*$ を選択し，$t_G$ を鍵とした $c$ の復号文を $\mathcal{C}^{-1}$ に求め，$R'$ を得る．その後，$m \in \{0,1\}^*, t_U, y_U \in_R \mathbb{Z}_q^*$ に対し，$m, t_U, y_U, t_G, c$ を $\mathcal{H}$ に質問し，$e$ を得る．このとき，Forking Lemma より無視できない確率で正しい署名 $(t_G, e, s_G)$ と $(t_G, \tilde{e}, \tilde{s}_G)$ を得る．このとき，$t_U y_U t_G \equiv g^{s_U+s_G}(y_U y_G)^e y_U \equiv g^{\tilde{s}_U+s_G}(y_U y_G)^{\tilde{e}} y_U \pmod{p}$ より，これらは $t_G = g^{s_G}(y_U y_G)^e = g^{\tilde{s_G}}(y_U y_G)^{\tilde{e}} \bmod p$ をみたすので，$g^{s_G-\tilde{s}_G} = (g^{x_U+x_G})^{e-\tilde{e}} \bmod p$ より，$x_G = (s_G - \tilde{s}_G)/(\tilde{e}-e) - x_U \bmod q$ を得る．これは，$y_G, g \in \mathbb{Z}_p^*$ に対する離散対数問題の解であり，離散対数問題の困難さの仮定に反する．従って，$InAdv$ が正当な署名を生成する確率は無視できる．

*Case 4.* $t_G \in_R \mathbb{Z}_p^*, c \in \{0,1\}^*$ を選択し，$t_G$ を鍵とした $c$ の暗号文を $\mathcal{C}$ に求め，$c'$ を得る．その後，$m \in \{0,1\}^*, t_U, y_U, y_G \in_R \mathbb{Z}_p^*$ に対し，$m, t, c$ を $\mathcal{H}$ に質問し，$e$ を得る．このとき，3 と同様の議論によって離散対数問題が解けることとなる．

*Case 1,2,3,4* より，$InAdv$ が検証式をみたす保証付署名を生成できる確率は無視できることがいえる．

## 4.4 ユーザに対する電子保証人方式

本節では，ユーザに対する電子保証を実現し，その安全性を考察する．この場合，保証者の生成する情報である保証書はユーザに対して生成され，ユーザの生成する情報はその保証書を用いて作成される．それによって，署名の正当性の検証が保証者の公開鍵なしではおこなうことができなくなる．

### 4.4.1 プロトコル

ユーザに対する電子保証方式を実現するためのプロトコルを以下で与える:

1. 初期設定: $q \mid p-1$ をみたす十分大きな素数 $p, q$ と $q$ を位数とする $g \in \mathbb{Z}_p^*$ を用意する．また，共通鍵暗号化関数，および共通鍵復号関数 $\mathcal{C}_K$，${\mathcal{C}_K}^{-1}$ と，ハッシュ関数 $\mathcal{H} : \{0,1\}^* \to \{0,1\}^{|n|}$ を用意する．

2. 鍵生成: 鍵生成アルゴリズム

$$KG_U(1^n) \ni (x_U, y_U), KG_G(1^n) \ni (x_G, y_G)$$

はセキュリティパラメータ $n$ に対し，$1^n$ の入力によって，鍵の組を出力する多項式時間アルゴリズムである．出力される $(x_U, y_U)$ と $(x_G, y_G)$ は，それぞれ $U$ と $G$ の秘密鍵と公開鍵の組である．ここで，$KG(1^n) \ni (x, y)$ は $(x, y)$ がアルゴリズム $KG$ への入力 $1^n$ によって出力されることを意味する．

– ユーザ $U$，保証者 $G$ ともに，秘密鍵として $x_U \in_R \mathbb{Z}_q^*, x_G \in_R \mathbb{Z}_q^*$ を選択し，$y_U := g^{x_U} \bmod p, y_G := g^{x_G} \bmod p$ をそれぞれ公開鍵とする．

3. 保証書生成: 保証書生成アルゴリズム

$$Gua(y_U, x_G, y_G) \ni \alpha_{UG}$$

は，ユーザ $U$ の公開鍵 $y_U$，保証者 $G$ の鍵の組 $(x_G, y_G)$ の入力に対して，$U$ への保証書 $\alpha_{UG}$ を出力する．

– 保証者 $G$ は，$r_G \in_R \mathbb{Z}_q^*$ を選択し，保証対象となるユーザの公開鍵 $y_U$ を用いて $t_G := g^{r_G} \bmod p, R_1 := 0^n \mid\mid {y_U}^{k_G}, R_2 := 0^n \mid\mid {y_U}^{k_G+1}$ を計算し，$t_G$ を鍵とした $R_1$ と $R_2$ の暗号文 $c_{G_1} := \mathcal{C}_{t_G}(R_1), c_{G_2} := \mathcal{C}_{t_G}(R_2)$ を求める．また，$e_G := \mathcal{H}(y_G, c_{G_1}, c_{G_2}), s_G := k_G - e_G x_G \bmod q$ とし，保証書 $\alpha_{UG} = (s_G, c_{G_1}, c_{G_2})$ を $U$ に送る．

4. 保証書を用いた署名生成: 署名生成アルゴリズム

$$Sig(m, x_U, y_U, y_G, \alpha_{UG}) \ni \sigma_{UG}$$

は，メッセージ $m$，ユーザ $U$ の鍵の組 $(x_U, y_U)$，保証者の公開鍵 $y_G$，保証書 $\alpha_{UG}$ の入力に対し，保証付署名 $\sigma_{UG}$ を出力する．

– ユーザ $U$ は，$r_U \in_R \mathbb{Z}_q^*$ を選択し，$\tilde{e}_G := \mathcal{H}(y_G, c_{G_1}, c_{G_2})$ に対して $t_{U_1} := g^{r_U} \bmod p, t_{U_2} := (g^{s_G} y_G{}^{\tilde{e}_G})^{r_U} \bmod p$ を計算し，$e_U := \mathcal{H}(m, t_{U_1} t_{U_2}), s_U := r_U - e_U x_U \bmod q$ とし，保証書付署名として $\sigma_{UG} = (e_U, s_U, s_G, (c_{G_1}, c_{G_2}))$ を出力する．

5-1. 一般検証: 一般検証アルゴリズム

$$Ver_{(U,G)}(m, \alpha_{UG}, \sigma_{UG}, y_U, y_G) \ni \{1, 0\}$$

は，メッセージ $m$，保証書 $\alpha_{UG}$，署名 $\sigma_{UG}$，公開鍵 $y_U, y_G$ に対し，$\sigma_{UG} \in Sig(m, x_U, y_U, y_G, \alpha_{UG})$ かつ，$\alpha_{UG} \in Gua(y_U, x_G, y_G)$ のときに圧倒的確率で 1 を，そうでないときは 0 を出力する．

– 検証者 $V$ は，$\tilde{e}_G := \mathcal{H}(y_G, c_{G_1}, c_{G_2})$ に対し，$\tilde{t}_G := g^{s_G} y_G{}^{\tilde{e}_G} \bmod p$ を計算する．$\tilde{t}_G$ を鍵として $c_{G_1}$ の復号文 $\tilde{R}_1 := \mathcal{C}_{\tilde{t}_G}{}^{-1}(c_{G_1})$ を求め，$\tilde{R}_1$ の上位 $n$ ビットを $b$ とし，$b||\tilde{\beta}$ に分割する．もし，$\tilde{t} := (g t_G)^{s_U} (y_U \beta)^{e_U} \bmod p$ に対して，$b = 0^n$ かつ $e_U = \mathcal{H}(m, \tilde{t})$ が成り立つならば，$\sigma_{UG}$ を $U$ の $G$ の保証付署名として受理し，そうでない場合は拒否する．

5-2. 保証書単独検証: 保証書単独検証アルゴリズム

$$Ver_G(m, \alpha_{UG}, \sigma_{UG}, y_G) \ni \{1, 0\}$$

は，メッセージ $m$，保証書 $\alpha_{UG}$，署名 $\sigma_{UG}$，保証者の公開鍵 $y_G$ に対し，$\sigma_{UG} \in Sig(m, x_U, y_U, y_G, \alpha_{UG})$ かつ，$\sigma_{UG} \in Gua(y_U, x_G, y_G)$ のときに圧倒的確率で 1 を，そうでないときは 0 を出力する．

– 検証者 $V$ は，$\tilde{e}_G := \mathcal{H}(y_G, c_{G_1}, c_{G_2})$ に対し，$\tilde{t}_G := g^{s_G} y_G{}^{\tilde{e}_G} \bmod p$ を計算する．$\tilde{t}_G$ を鍵として $c_{G_2}$ の復号文 $\tilde{R}_2 := \mathcal{C}_{\tilde{t}_G}{}^{-1}(c_{G_2})$ を求め，$\tilde{R}_2$ の上位 $n$ ビットを $b$ とし，$b||\tilde{\beta}$ に分割する．もし，$\tilde{t} := (g t_G)^{s_U} \beta^{e_U} \bmod p$ に対して，$b = 0^n$ かつ $e_U = \mathcal{H}(m, \tilde{t})$ が成り立つならば，$\sigma_{UG}$ を $G$ の保証付署名として受理し，そうでない場合は拒否する．

### 4.4.2 安全性の考察

ここで，ユーザに対する電子保証人方式の安全性について考察する．システム・パラメータ，公開鍵が与えられ，ランダム・オラクル $\mathcal{H}$ と理想的共通鍵暗号関数 $\mathcal{C}$，または復号関数 $\mathcal{C}^{-1}$ にアクセスを許された攻撃者 $Adv$ を仮定する．このとき，ユーザ，及び保証者に対する安全性について以下の定理が与えられる．

**定理 3** (保証者に対する安全性) ユーザ $U$ と結託することによって，適応的に選択した公開鍵 $y_{U_i}$ $(1 \leq i \leq \ell)$ に対応する正当な保証書 $\alpha_{U_iG}$ を得ることのできる攻撃者 $Adv$ が，$y_U \neq \forall y_{U_i}$ に対して

$$Ver_{(U,G)}(m, \alpha_{UG}, \sigma_{UG}, y_U, y_G) = 1 \vee Ver_G(m, \alpha_{UG}, \sigma_{UG}, y_G)$$

をみたす保証書 $\alpha_{UG}$ を生成できる確率は無視できる．

*Proof.* 攻撃者 $Adv$ の動きを以下の 4 つに分類し，それぞれについて考察する．

*Case 1.* $y_U, t_G \in_R \mathbb{Z}_p^*, c_1, c_2 \in_R \{0,1\}^*$ を $\mathcal{H}$ に質問し，答えとして $e_G$ を得た後，$t_G$ を鍵とした $R_1, R_2$ の暗号文を $\mathcal{C}$ に求め，$c_1' = \mathcal{C}_{t_G}(R_1)$，及び $c_2' = \mathcal{C}_{t_G}(R_2)$ を得る．このとき，既に選択した $c_1, c_2$ に対して $c_1 = c_1'$ または $c_2 = c_2'$ となる確率は無視できるほど小さい．

*Case 2.* $y_U, t_G \in_R \mathbb{Z}_p^*, c_1, c_2 \in_R \{0,1\}^*$ を $\mathcal{H}$ に質問し，答えとして $e_G$ を得た後，$t_G$ を鍵とした $c_1, c_2$ の復号文を $\mathcal{C}^{-1}$ を求め，$R_1' = {\mathcal{C}_{t_G}}^{-1}(c_1)$，及び $R_2' = {\mathcal{C}_{t_G}}^{-1}(c_2)$ を得る．このとき，$R_1$ または $R_2$ が $n$ ビットの冗長をもつ，即ち $0^n \mid\mid \beta$ という形になる確率は無視できるほど小さい．

*Case 3.* $t_G \in_R \mathbb{Z}_q^*, c_1, c_2 \in \{0,1\}^*$ を選択し，$t_G$ を鍵とした $c_1, c_2$ の復号文を $\mathcal{C}^{-1}$ に求め，$R_1'$，及び $R_2'$ を得る．その後，$m \in \{0,1\}^*$，$t_U, y_U \in_R \mathbb{Z}_q^*, c_1, c_2 \in \{0,1\}^*$ を $\mathcal{H}$ に質問し，$e_G$ を得る．このとき，Forking Lemma より無視できない確率で正しい保証書 $(t_G, e_G, s_G)$ と $(t_G, \tilde{e}_G, \tilde{s}_G)$ を得る．このとき，$t_G \equiv g^{s_G} {y_G}^{e_G} \equiv g^{\tilde{s}_G} {y_G}^{\tilde{e}_G} \pmod{p}$ より，$g^{s_G - \tilde{s}_G} \equiv {y_G}^{\tilde{e}_G - e_G} \pmod{p}$ を得る．これより $x_G \equiv (s_G - \tilde{s}_G)/(\tilde{e}_G - e_G) \pmod{q}$ を得るが，これは $y_G, g \in \mathbb{Z}_p^*$ に対する離散対数問題の解であり，離散対数問題の困難さの仮定に反する．従って，$Adv$ が正当な署名を生成する確率は無視できる．

*Case 4.* $t_G \in_R \mathbb{Z}_p^*, R_1, R_2 \in \{0,1\}^*$ を選択し，$t_G$ を鍵とした $R_1, R_2$ の暗号文を $\mathcal{C}$ に求め，$c_1, c_2$ を得る．その後，$y_U \in_R \mathbb{Z}_p^*$ に対し，$m, t, c$ を $\mathcal{H}$ に質問し，$e'_G$ を得る．このとき，3 と同様の議論によって離散対数問題が解けることとなる．

*Case 1,2,3,4* より，$Adv$ が検証式をみたす保証付署名を生成できる確率は無視できることがいえる．

**定理 4** (署名者に対する安全性) 保証者 $G$ と結託することによって，適応的に選択したメッセージ $m_i$ $(1 \leq i \leq \ell)$ に対する $U_i$ の署名 $\sigma_{U_iG}(m_i)$ を得ることができる攻撃者 $Adv$ が $(m, U) \neq \forall(m_i, U_i)$ に対して

$$Ver_{(U,G)}(m, \alpha_{UG}, \sigma_{UG}, y_U, y_G) = 1 \vee Ver_G(m, \alpha_{UG}, \sigma_{UG}, y_G) = 1$$

をみたす保証付署名 $\sigma_{UG}$ を生成できる確率は無視できる．

*Proof.* ランダム・オラクル仮定の下，$Adv$ がメッセージ $m$ に対する $U$ の署名 $\sigma_{UG}(m)$ を偽造できたと仮定する．このとき，Forking Lemma より $Adv$ は $U$ の署名 $(t, e_U, s_U)$ と $(t, \tilde{e}_U, \tilde{s_U})$ を得ることができ，$t \equiv (g^{k_G+1})^{s_U}({y_U}^{k_G+1})^{e_U} \equiv (g^{k_G+1})^{\tilde{s}_U}({y_U}^{k_G+1})^{\tilde{e}_U}$ より，$x_U = (s_U - \tilde{s}_U)/(e_U - \tilde{e_U}) \bmod q$ を得る．これは，$y_U, g \in \mathbb{Z}_p^*$ に対する離散対数問題の解であり，離散対数問題の困難さの仮定に反する．従って，$Adv$ が正当な署名を生成する確率は無視できる．

## 4.5 考察

ここで，提案手法の効率性について考察をおこなう．$\mathbb{Z}_p$ 上での 1 回の剰余乗算に必要な計算コストを $M(p)$，1 回の共通鍵暗号の復号に必要なコストを $C$ とし，剰余べき演算にバイナリ法を用いたそれぞれの検証コストを表 4.1 にまとめる．

提案方式である取引に対する電子保証において，一般検証に必要となる計算コストは $(7|q|/4)M(p)$，保証書単独検証に必要となる計算コストは $(7|q|/2)M(p) + C$ となる．一般検証においては，2 者による多重署名より小さい計算コストで実現できている．実際，ディジタル署名の検証は，公開鍵証明書発行機関 (CA) によって発行される公開鍵証明書の正当性を検証したうえでおこなう必要があり，公開鍵証明書が Schnorr 署名で生成されている場合，$U$ と $G$ の公開鍵の検証にそれぞ

れ $(7|q|/2)M(p)$ が必要となるため，一般検証には全体で $(21|q|/4)M(p)$ の計算コストが必要となる．また，保証単独検証は保証者の公開鍵検証のみを必要とするため，$(7|q|/4)M(p)$ の計算を加えた $(21|q|/4)M(p)+C$ の計算が必要となる．よって，実際の保証書単独検証は，一般検証に必要な計算コストに加え，1 回の復号処理を必要とするが，ユーザの公開鍵を参照するための通信コストを必要としないため，検証者への負担を軽くすることができると考えられる．また，ユーザに対する電子保証においても，公開鍵証明書の適用とほぼ同等の計算コストで実現することができている．

表 4.1: 電子保証人方式における効率性の比較

| 方式 | 一般検証 | 保証書単独検証 | 満たさない性質 |
|---|---|---|---|
| 多重署名 [39] | $(15\|q\|/8)M(p)$ | — | 保証単独検証可能性<br>署名単独検証不可能性 |
| 提案方式 (取引保証) | $(7\|q\|/4)M(p)$ | $(7\|q\|/2)M(p)+C$ | |
| 公開鍵証明書 [47] | $(7\|q\|/2)M(p)$ | $(7\|q\|/4)M(p)$ | 署名単独検証不可能性 |
| 提案手法 (ユーザ保証) | $(7\|q\|/2)M(p)+C$ | $(7\|q\|/2)M(p)+C$ | |

# 第 5章

# 匿名証明書方式

## 5.1 はじめに

近年のオンライン・システムの普及に伴い，個人情報の拡散が問題となっている．このため，ユーザのプライバシ確保のためには，個人が自らの情報の流通を制御できるアプリケーションが必要であり，このような問題を解決するのが匿名証明書方式 [15, 17, 18, 7, 32, 11] である．匿名証明書方式では，ユーザが異なる仮名 (Pseudonym) を取引機関へ登録する．各機関はその仮名によってユーザの情報を管理・識別するため，それぞれの機関のもつ情報のリンクによる個人情報の拡散を防ぐことができ，機関との取引は，登録した仮名に対応するユーザであることを証明することでおこなわれる．また，各機関は登録されたユーザの仮名に対する登録証明書 (Credential) を発行し，対応するユーザは証明書の保持証明をおこなうことで，別機関にその機関への登録者であるという事実のみを示すことができる．このとき，仮名に関する情報を検証者である別機関に何ら与えないため，登録証明による仮名の関連付けを防ぐことができる．ここで，匿名証明書方式に必要な性質を挙げる．

ユーザの匿名性: 検証者，および検証機関はそのユーザの証明書保持の事実以外，ユーザに関する情報を何も得ることはできない．

仮名の関連付け不可能性: 同一ユーザの異なる仮名が関連付けされることはない．

匿名証明書の偽造不可能性: 機関によって発行される登録証明書の偽造をおこなう

ことはできない.

## 5.2 準備

### 5.2.1 理想的な匿名証明書方式

ここで，準備として理想的な匿名証明書方式の定義を与える [11]．理想的なシステムは信頼できる第三者 $T$ の仲介によって構築され，全てのやりとりは $T$ を介して行われる．また，$T$ は他のエンティティに対するユーザの匿名性を保証する．ここで，理想的な匿名証明書方式 $ICS$ と中間者 $T$ を用いない暗号学を用いて構築された匿名証明書方式 $CCS$ に対して，安全なシステムの定義を以下のように与える:

**定義 12** セキュリティ・パラメータ $k$ に対する $V = poly(k)$ をシステム内のエンティティの数とする．セキュリティパラメータ $k$，システム内で実行されるイベントに対するイベントスケジューラ $E$ に対する理想的匿名証明書方式を $ICS(1^k, E)$，暗号学を用いて構築された匿名証明書方式を $CCS(1^k, E)$ とする．また，システム内のエンティティ $i$ の出力を $Z_i(1^k)$ であらわし，システム内に存在するすべてのエンティティの出力を $\{A_1(1^k), \ldots, A_V(1^k)\}$，匿名証明書方式 $CS$ 内で得られる出力を $\{A_1(1^k), \ldots, A_V(1^k)\}^{CS(1^k,E)}$ と記述することにする．

このとき，全ての確率的多項式チューリング・マシン $\mathcal{A}$ と十分大きな $k$ に対して，以下のような性質をみたすシミュレータ $\mathcal{S}$ が存在するならば $CCS$ は安全であるという:
(1) $ICS$ において，$\mathcal{S}$ は $\mathcal{A}$ によってコントロールされた実世界のエンティティに対応する理想世界のエンティティをコントロールする．
(2) 全てのイベントスケジューラ $E^{\mathcal{A}}$ に対して，$\mathcal{S}$ は $\mathcal{A}$ にアクセスするブラックボックスを用いることができ，

$$\{\{Z_i(1^k)\}_{i=1}^{V}, \mathcal{A}(1^k)\}^{CCS(1^k,E)} \stackrel{c}{\approx} \{\{Z_i(1^k)\}_{i=1}^{V}, \mathcal{S}^{\mathcal{A}}(1^k)\}^{ICS(1^k,E)}.$$

をみたす．但し，$D_1(1^k) \stackrel{c}{\approx} D_2(1^k)$ は $D_1$ と $D_2$ が計算量的に識別不可能であることを示す．

## 5.3 匿名性を強化した証明書方式

本節では，匿名性を強化した証明書方式を与える．既存の方式 [15, 17, 18, 7, 32, 11] では，システム内にユーザと機関のみが存在し，各機関はユーザに登録証明書を発行する．よって，登録証明書の保持証明に機関の公開鍵を必要とすることから，ユーザに強い匿名性をもたせることができなかった．つまり，仮名と登録している機関とが，証明書から関連付けされていた．そこで，提案する匿名証明書方式では，機関を適当にグループ化し，グループ内の機関管理者が機関情報を埋め込んだ登録証明書をユーザに発行する．これによって，ユーザは埋め込まれた機関情報の露出を制御することができ，自らのセキュリティ・ポリシーに応じた登録証明をおこなうことができる．即ち，ユーザはグループに属すること，もしくはグループ内のある機関に属することのいづれかの証明を選択することができる．また，提案する手法は，既存方式のもつ性質である，ユーザの匿名性，仮名の関連付け不可能性，および登録証明書の偽造不可能性をもちあわせている．

提案する匿名証明書方式は以下のエンティティによって構成される:

グループ $\boldsymbol{G}$: 機関の集合.

機関管理者 $M_G$: グループ $\boldsymbol{G}$ の秘密鍵を管理し，その秘密鍵を用いてユーザに証明書を発行するエンティティ.

機関 $O_i$: グループに属するエンティティ.

ユーザ $U$: グループに登録し，そのグループ内の機関と取引をおこなうエンティティ.

検証者 $V$: ユーザの属性を検証するエンティティ.

### 5.3.1 提案方式の概要

ここで，匿名性を強化した証明書方式の概要を述べる．提案する手法は以下のように構築される．

但し，ユーザ，各機関，機関管理者間は匿名通信路で結ばれているものとする．

1. 初期設定:

   全ての機関管理者 $M_G \in \boldsymbol{G}$ は，鍵生成プロトコル $KG_G$ を用いてグループの秘密鍵，公開鍵の組を生成する．$M_G$ によって生成されるグループ $\boldsymbol{G}$ の公開鍵を RSA の法 $n_G$ と $d_G, e_G, f_G, g_G, h_G, v_G \in {\mathbb{Z}^*_{n_G}}^2$ とし，秘密鍵を $n_G$ の素因数とする．

2. 機関 $O_i$ のグループ $\boldsymbol{G}$ への登録:

   機関 $O_i$ は，グループ $\boldsymbol{G}$ へ登録するためにグループの公開鍵を用いて鍵生成プロトコル $KG_{(O_i,G)}$ から，秘密鍵と公開鍵の組 $(x_{(O_i,G)}, y_{(O_i,G)} := {v_G}^{x_{(O_i,G)}} \bmod n_G)$ を生成し，識別情報 $id_{(O_i,G)}$ とともにグループに登録する．機関管理者 $M_G$ は登録された機関の公開鍵のリストを公開する．

3. ユーザの登録:

   3-1. ユーザ $U$ のグループ $\boldsymbol{G}$ への登録:

   ユーザ $U$ は，鍵生成プロトコル $KG_U$ を用いてシステム内で用いる秘密鍵 $x_U$ を生成する．また，仮名生成プロトコル $PG$ によって，仮名 $P_{(U,G)}$ を生成し，グループ $\boldsymbol{G}$ に登録する．これは $U$ の秘密鍵 $x_U$ を用いて $P_{(U,G)} := {g_G}^{x_U} {h_G}^{s_{(U,G)}} \bmod n_G$ で生成される．ただし，$s_{(U,G)}$ はプロトコル $PG$ によって生成される $U$ の秘密情報である．その後，機関管理者 $M_G$ はグループ登録証明書発行プロトコル $CI_G$ によって登録証明書を発行する．この証明書は $C_{(U,G)} \equiv (P_{(U,G)} f_G)^{1/E_{(U,G)}} \pmod{n_G}$ をみたす $(E_{(U,G)}, C_{(U,G)})$ の組であらわされる．

   3-2. ユーザ $U$ の機関 $O_i \in \boldsymbol{G}$ への登録:

   **i.** ユーザ $U$ は機関 $O_i$ に仮名生成プロトコル $PG$ を用いて生成した $P_{(U,O_i)} := {g_G}^{x_U} {h_G}^{x_{(U,O_i)}} \bmod n_G$ を登録する．

   **ii.** $U$ は検証機関を $O_i$ としたグループ登録証明プロトコル $CT_G$ によってグループ $\boldsymbol{G}$ への登録証明をおこなうとともに，登録した仮名 $P_{(U,O_i)}$ が正しい型であること，即ち $U$ の秘密鍵を用いて生成されていることを証明する．

   **iii.** $U$ と $O_i$ は仮名保証書生成プロトコル $GG$ によって $U$ の $O_i$ への登録を機関管理者に保証するための情報である $\sigma_{(U,O_i)}$ を生成する．

$\sigma_{(U,O_i)}$ は仮名 $P_{(U,O_i)}$ に対して生成され，$U$ へのプライベート出力である．

**iv.** $U$ は機関管理者 $M_G$ に機関登録証明書発行プロトコル $CI_O$ によって $\sigma_{(U,O_i)}$ の正当性を証明し，$M_G$ は機関 $O_i$ の登録証明書を発行する．$\sigma_{(U,O_i)}$ の正当性とは，$\sigma_{(U,O_i)}$ が機関 $O_i$ によって生成されたものであり，かつ $P_{(U,G)}$ と同じ秘密鍵を用いて生成された仮名に対して発行されたものであることを意味する．機関登録証明書は $O_i$ の識別情報である $id_{(O_i,G)}$ を用いて生成され，$C_{(U,O_i)} \equiv (P_{(U,O_i)} {d_G}^{id_{(O_i,G)}} f_G)^{1/E_{(U,O_i)}} \pmod{n_G}$ をみたす $(E_{(U,O_i)}, C_{(U,O_i)})$ の組であらわされる．

4. 登録証明: ユーザ $U$ は検証者 $V$ もしくは検証機関 $O_j \in \boldsymbol{G}_J$ に登録証明をおこなう．

4-1. 検証者への $O_i$ への登録，またはグループ $\boldsymbol{G}_I$ への登録証明: もし，検証者 $V$ に $O_i \in \boldsymbol{G}_I$ への登録を示す場合は，機関登録証明プロトコル $CS^+$ によって $O_i \in \boldsymbol{G}_I$ への登録を証明する．また，検証者 $V$ にその登録を示したくない，もしくはその必要がない場合には機関情報をもたない機関登録証明プロトコル $CS^-$ によってグループ $\boldsymbol{G}_I$ に属するある機関の登録証明をおこなうことができる．

プロトコル $CS^+$ では，$id_{(O_i,G)}$ を検証者に与え，

$$ {C_{(U,O_i)}}^{E_{(U,O_i)}} \equiv {g_G}^{x_U} {h_G}^{s_{(U,O_i)}} {d_G}^{id_{(O_i,G)}} f_G \pmod{n_G} $$

をみたす，$C_{(U,O_i)}, E_{(U,O_i)}, x_U, s_{(U,O_i)}$ の知識証明をおこない，プロトコル $CS^-$ では $id_{(O_i,G)}$ を検証者に与えず，$C_{(U,O_i)}, E_{(U,O_i)}, x_U, s_{(U,O_i)}$ に加えて $id_{(O_i,G)}$ の知識証明をおこなう．

4-2. 検証機関への $O_i$ への登録，またはグループ $\boldsymbol{G}_I$ への登録証明: ユーザ $U$ は検証機関 $O_j \in \boldsymbol{G}_J$ に機関登録証明プロトコル $CT^+$ によって $O_i \in \boldsymbol{G}_I$ への登録証明書をおこなうとともに $P_{(U,O_j)}$ に対応するユーザであることを証明する．また，$U$ がその登録の事実を $O_j$ に知らせたくない場合には，グループ $\boldsymbol{G}_I$ に属するある機関への登録証明とともに，$P_{(U,O_j)}$ に対応するユーザであることが証明できる．

プロトコル $CT^+$ では，$id_{(O_i,G)}$ を検証者に与え，

$$C_{(U,O_i)}{}^{E_{(U,O_i)}} \equiv {g_{G_I}}^{x_U} {h_{G_I}}^{s_{(U,O_i)}} {d_{G_I}}^{id_{(O_i,G)}} f_{G_I} \pmod{n_{G_I}}$$

$$P_{(U,O_j)} \equiv {g_{G_J}}^{x_U} {h_{G_J}}^{s_{(U,O_j)}} \pmod{n_{G_J}}$$

を同時にみたす，$C_{(U,O_i)}, E_{(U,O_i)}, x_U, s_{(U,O_i)}, s_{(U,O_j)}$ の知識証明をおこなうことで，$P_{(U,O_j)}$ に対応するユーザの $O_i$ への登録を示すことができ，プロトコル $CT^-$ では $id_{(O_i,G)}$ を検証者に与えず，$C_{(U,O_i)}, E_{(U,O_i)}, x_U, s_{(U,O_i)}, s_{(U,O_j)}$ に加えて $id_{(O_i,G)}$ の知識証明をおこなう．

### 5.3.2 プロトコル

本節では，匿名性を強化した匿名証明書方式を提案する．

#### 初期設定

システムパラメータを以下のように設定する．RSA の法のサイズを $\ell_n$ ビット，セキュリティパラメータを $\epsilon > 1$ する．また，$\ell_\Gamma = 2\ell_n$, $\ell_\Delta = \epsilon\ell_\Gamma$, $2^{\ell_\Lambda} > 2(2^{2\ell_\Gamma} + 2^{\ell_\Gamma} + 2^{2\ell_\Delta} + 2^{\ell_\Delta})$ とし，各区間を $\Gamma = ]-2^{\ell_\Gamma}, 2^{\ell_\Gamma}[$, $\Delta = ]-2^{\ell_\Delta}, 2^{\ell_\Delta}[$, $\Lambda = ]2^{\ell_\Lambda}, 2^{\ell_\Lambda+\ell_\Sigma}[$ と設定する．

#### 鍵生成プロトコル $KG$

鍵生成プロトコル $KG_G, KG_{(O,G)}$ と $KG_U$ は

$$KG_G(1^k) \ni (\mathcal{X}_G, \mathcal{Y}_G), KG_{(O,G)}(1^k, \mathcal{Y}_G) \ni (x_{(O,G)}, y_{(O,G)}), KG_U(1^k) \ni x_U$$

であらわされ，$KG_G$ と $KG_{(O,G)}$ は，それぞれグループ $\boldsymbol{G}$ と機関 $O \in \boldsymbol{G}$ の秘密鍵，公開鍵の組を，$KG_U$ はユーザ $U$ の秘密鍵を生成する．$(\mathcal{X}_G, \mathcal{Y}_G)$ をグループ $\boldsymbol{G}$ の鍵，$(x_{(O,G)}, y_{(O,G)})$ を機関 $O \in \boldsymbol{G}$ の鍵，そして $x_U$ をユーザ $U$ の秘密鍵とする．上記は，$KG_G$ と $KG_U$ は $1^k$ の入力に対し $(\mathcal{X}_G, \mathcal{Y}_G)$ または $x_U$ を，また $KG_{(O,G)}$ は $1^k$ とグループの公開鍵 $\mathcal{Y}_G$ に対し $(x_{(O,G)}, y_{(O,G)})$ を出力することを意味する．

*Step 1.* 機関管理者 $M_G \in \boldsymbol{G}$ は，$p_G := 2p'_G + 1$ かつ $q_G := 2q'_G + 1$ が素数となるような $\ell_n/2$ ビットの素数 $p'_G, q'_G$ を選択し，$n_G := p_G q_G$ を計算する．また，$d_G, e_G, f_G, g_G, h_G, v_G \in_R {\mathbb{Z}^*_{n_G}}^2$ を選択し，$\mathcal{X}_G := (p_G, q_G)$ をグループの秘密鍵，$\mathcal{Y}_G := (n_G, d_G, e_G, f_G, g_G, h_G, v_G)$ をグループの公開鍵とする．

*Step 2.* 各組織 $O_i$ は，$x_{(O_i,G)} \in_R \Gamma$ に対して $y_{(O_i,G)} := {g_G}^{x_{(O_i,G)}} \bmod n_G$ を計算し，$y_{(O_i,G)}$ をグループ $\boldsymbol{G}$ への登録鍵とする．

*Step 3.* ユーザ $U$ は，システム内で用いる秘密鍵として $x_U \in \Gamma$ を選択し，保管しておく．

**仮名生成プロトコル** $PG$

仮名生成プロトコル $PG$ は，

$$PG\langle U(x_U), X\rangle(\mathcal{Y}_G) \ni [U(s_{(U,X)})](P_{(U,X)})$$

であらわされる．上記は，$PG$ は $U$ と $X \in \{O \in \boldsymbol{G}, M_G \in \boldsymbol{G}\}$ の対話によっておこなわれ，これは，$U$ による入力 $x_U$ と共通入力 $\mathcal{Y}_G$ から，$U$ へ $s_{(U,X)}$ をプライベート出力し，$P_{(U,X)}$ を $U$ と $X$ へ共通出力することを意味する．また，$PG$ は出力 $P_{(U,X)}$ が正しい型であること，即ち $x_U \in \Gamma, s_{(U,X)} \in \Delta$ を用いて $P_{(U,X)} = {g_G}^{x_U} {h_G}^{s_{(U,X)}}$ にあらわされることを保証する．

プロトコル $PG$ は以下のように構成される:

*Step 1.* $U$ は $r_1 \in_R \Delta, r_2, r_3 \in_R \{0,1\}^{2\ell_n}$ に対し，$c_1 := {d_G}^{r_1} {e_G}^{r_2}, c_2 := {d_G}^{x_U} {e_G}^{r_3}$ を計算し，それらを $X$ に送る．

$$PK^2\{(\alpha, \beta, \gamma, \delta) \ : \ c_1 = {d_G}^{\alpha} {e_G}^{\beta} \wedge c_2 = {d_G}^{\gamma} {e_G}^{\delta}\}$$

によって，$c_1, c_2$ が正しく生成されていることを証明する．

*Step 2.* $X$ は $r \in_R \Delta$ を選択し，$U$ に送信する．

*Step 3.* $U$ は，$s_{(U,X)} := (r_1 + r \pmod{2^{\ell_\Delta + 1} + 1}) - 2^{\ell_\Delta} + 1$ と $\tilde{s} = \left\lfloor \frac{r_1 + r}{2^{\ell_\Delta + 1} - 1} \right\rfloor$ を計算し，$P_{(U,X)} := {g_G}^{x_U} {h_G}^{s_{(U,X)}}$ を仮名とする．また $r_4 \in_R \{0,1\}^{\ell_n}$ に対し

$c_3 := {d_G}^{\tilde{s}} {e_G}^{r_4}$ と $P_{(U,X)}$ を $X$ に送信し，それが正しく生成されたことを以下の $PK^2$ によって証明する:

$$\begin{aligned} PK^2\{(\alpha,\beta,\gamma,\delta,\varepsilon,\zeta,\vartheta,\xi) \quad : \quad & c_1 = {d_G}^{\alpha} {e_G}^{\beta} \\ \wedge \quad & c_2 = {d_G}^{\gamma} {e_G}^{\delta} \\ \wedge \quad & c_3 = {d_G}^{\varepsilon} {e_G}^{\zeta} \\ \wedge \quad & P_{(U,X)} = {g_G}^{\gamma} {h_G}^{\vartheta} \\ \wedge \quad & (c_1 {d_G}^{r-2^{\ell_\Delta+1}})/({c_3}^{2^{\ell_\Delta+1}+1}) = {d_G}^{\vartheta} {e_G}^{\xi} \\ \wedge \quad & \gamma \in \Gamma \wedge \vartheta \in \Delta\}. \end{aligned}$$

*Step 4.* $X$ は仮名リストに $P_{(U,X)}$ を保管する．

*Step 5.* $U$ は $P_{(U,X)}$ の生成に用いた秘密情報 $s_{(U,X)}$ と $P_{(U,X)}$ を $X$ の登録情報として秘密に保管する．

### グループ登録証明書発行プロトコル $CI_G$

$CI_G$ は，機関管理者 $M_G \in \boldsymbol{G}$ がグループ登録証明書を発行するのに用いられ，

$$CI_G\langle U(x_U, s_{(U,G)}), M_G(\mathcal{X}_G)\rangle(\mathcal{Y}_G, P_{(U,G)}) \ni \mathcal{C}_{(U,G)}$$

であらわされる．$CI_G$ は $U$ によるプライベート入力 $x_U, s_{(U,G)}$ と，$M_G$ によるプライベート入力 $\mathcal{X}_G$，共通入力 $\mathcal{Y}_G, P_{(U,G)}$ に対して，$P_{(U,G)} \in PG\langle U, M_G\rangle$ に対する証明書 $\mathcal{C}_{(U,G)}$ を出力する．この証明書 $\mathcal{C}_{(U,G)}$ は仮名 $P_{(U,G)}$ に対して $C_{(U,G)} \equiv (P_{(U,G)} f_G)^{1/E_{(U,G)}} \pmod{n_G}$ をみたす $(E_{(U,G)}, C_{(U,G)})$ の組であらわされる．

プロトコル $CI_G$ は，以下のように構成される:

*Step 1.* ユーザ $U$ は $PK^2\{(\alpha,\beta) : P_{(U,G)} = {g_G}^{\alpha} {h_G}^{\beta}\}$ によって $M_G$ のデータベースに登録されている $P_{(U,G)}$ に対応するユーザであることを証明する．

*Step 2.* 機関管理者 $M_G$ は素数 $E_{(U,G)} \in_R \Lambda$ を選択し，$C_{(U,G)} := (P_{(U,G)} f_G)^{1/E_{(U,G)}} \bmod n_G$ を計算する．また $U$ にグループ登録証明書として $(E_{(U,G)}, C_{(U,G)})$ を送る．機関管理者 $M_G$ は $\mathcal{C}_{(U,G)} = (E_{(U,G)}, C_{(U,G)})$ をそれに対応する仮名 $P_{(U,G)}$ とともに保管する．

*Step 3.* $U$ は ${C_{(U,G)}}^{E_{(U,G)}} \equiv P_{(U,G)} f_G \pmod{n_G}$ かつ $E_{(U,G)} \in \Lambda$ であるかどうかを検証し，正しければ $\mathcal{C}_{(U,G)} = (E_{(U,G)}, C_{(U,G)})$ をグループ $\boldsymbol{G}$ の登録証明書として保管する．

### グループ登録証明プロトコル $CS_G$

$CS_G$ は，ユーザ $U$ が検証者 $V$ にグループ $\boldsymbol{G}$ への登録を証明するのに用いられる．これは

$$CS_G\langle U(\mathcal{C}_{(U,G)}, x_U, s_{(U,G)}), V\rangle(\mathcal{Y}_G) \ni \{0,1\}$$

であらわされ，証明者 $U$ が $\mathcal{C}_{(U,G)} \in CI_G\langle U, M_G\rangle$ を保持するとき圧倒的確率で 1 を，そうでないときは 0 を出力する．

プロトコル $CS_G$ は以下のように構成される:

*Step 1.* $U$ は $r_1, r_2 \in_R \{0,1\}^{2\ell_n}$ を選択し，$c_1 := C_{(U,G)} {e_G}^{r_1}, c_2 := {e_G}^{r_1} {d_G}^{r_2}$ を計算し，検証者 $V$ に $c_1, c_2$ を送る．

*Step 2.* $U$ は以下の $PK^2$ によって検証者 $V$ に登録証明をおこなう:

$$\begin{aligned} PK^2\{(\alpha,\beta,\gamma,\delta,\varepsilon,\zeta,\xi) \quad : \quad & f_G = {c_1}^{\alpha} / {g_G}^{\beta} {h_G}^{\gamma} {e_G}^{\delta} \\ \wedge \quad & c_2 = {e_G}^{\varepsilon} {d_G}^{\zeta} \\ \wedge \quad & 1 = {c_2}^{\alpha} / {e_G}^{\delta} {d_G}^{\xi} \\ \wedge \quad & \alpha \in \Lambda \wedge \beta \in \Gamma \wedge \gamma \in \Delta\}. \end{aligned}$$

### 検証機関に対するグループ登録証明プロトコル $CT_G$

$CT_G$ は，ユーザ $U$ がエンティティ $X \in \{O_j \in \boldsymbol{G}_J, M_{G_J} \in \boldsymbol{G}_J\}$ にグループ $\boldsymbol{G}_I$ の登録を示すとともに，$X$ における $P_{(U,X)}$ に対応するユーザであることを証明するのに用いられる．

$$CT_G\langle U(x_U, s_{(U,G_I)}, s_{(U,G_J)}, P_{(U,G_I)}, \mathcal{C}_{(U,G_I)}), X\rangle(\mathcal{Y}_{G_I}, \mathcal{Y}_{G_J}, P_{(U,X)}) \ni \{0,1\}$$

であらわされ，証明者 $U$ が $\mathcal{C}_{(U,G_I)} \in CI_G\langle U, M_{G_I}\rangle(P_{(U,G_I)}), P_{(U,G_I)} \in PG\langle U(x_U), M_{G_I}\rangle$ かつ $P_{(U,X)} \in PG\langle U(x_U), X\rangle$ をみたす $\mathcal{C}_{(U,G_I)}, P_{(U,G_I)}$ を入力したとき，共通入力 $P_{(U,X)}$ に対して圧倒的確率で 1 を，そうでないときは 0 を出力する．

プロトコル $CT_G$ は以下のように構成される:

*Step 1.* ユーザ $U$ は $r_1, r_2 \in_R \{0,1\}^{2\ell_n}$ を選択し，$c_1 := C_{(U,G_I)} {e_G}^{r_1}$ と $c_2 := {e_{G_I}}^{r_1} {d_{G_I}}^{r_2}$ を計算し，$c_1, c_2$ を検証機関 $X$ に送る．

*Step 2.* $U$ は以下の $PK^2$ によって，検証機関 $X$ に登録証明をおこなう:

$$\begin{aligned} PK^2\{(\alpha,\beta,\gamma,\delta,\varepsilon,\zeta,\xi,\eta) \quad : \quad & f_{G_I} = {c_1}^{\alpha} / {g_{G_I}}^{\beta} {h_{G_I}}^{\gamma} {e_{G_I}}^{\delta} \\ \wedge \quad & c_2 = {e_{G_I}}^{\varepsilon} {d_{G_I}}^{\zeta} \\ \wedge \quad & 1 = {c_2}^{\alpha} / {e_{G_I}}^{\delta} {d_{G_I}}^{\xi} \\ \wedge \quad & P_{(U,X)} = {g_{G_J}}^{\beta} {h_{G_J}}^{\eta} \\ \wedge \quad & \alpha \in \Lambda \wedge \beta \in \Gamma \wedge \gamma \in \Delta\}. \end{aligned}$$

**仮名保証書生成プロトコル** $GG$

$GG$ によって，ユーザ $U$ は仮名を登録した機関 $O_i \in \boldsymbol{G}$ から，登録済みであることを保証する情報を得ることができる．このプロトコルは，

$$GG\langle U(x_U, s_{(U,O_i)}), O_i(x_{(O_i,G)})\rangle(\mathcal{Y}_G, y_{(O_i,G)}, P_{(U,O_i)}) \ni [U(\sigma_{(U,O_i)}, r_{(U,O_i)})]$$

であらわされ，$U$ のプライベート入力 $x_U$ に対し，$P_{(U,O_i)} \in PG\langle U(x_U), O_i\rangle$ のときのみ $U$ の仮名 $P_{(U,O_i)}$ に対する仮名保証書 $\sigma_{(U,O_i)} = (\tilde{e}, \tilde{s}, \tilde{P}, \tilde{Q})$ と乱数 $r_{(U,O_i)}$ をプライベート出力として $U$ に与える．

また，$I_i \in \boldsymbol{G}$ によって生成された正しい仮名保証書 $\sigma_{(U,O_i)}$ は，公開鍵 $y_{(O_i,G)}$ に対して，

$$\tilde{e} = \mathcal{H}(g_G, y_{(O_i,G)}, \tilde{P}, \tilde{Q}, {g_G}^{\tilde{s}} {y_{(U,O_i)}}^{\tilde{e}}, \tilde{P}^{\tilde{s}} \tilde{Q}^{\tilde{e}})$$

をみたす．プロトコル $GG$ は以下のように構成される:

*Step 1.* $U$ は $O_i$ に登録済みの仮名 $P_{(U,O_i)}$ に対応するユーザであることを

$$PK^2\{(\alpha,\beta) : P_{(U,O_i)} = {g_G}^{\alpha} {h_G}^{\beta}\}$$

によって証明する．

*Step 2.* $O_i$ は，$r \in_R \{0,1\}^{2\ell_n}$ に対して $t_1 := {g_G}^r, t_2 := {P_{(U,O_i)}}^r$ と $Q_{(U,O_i)} := {P_{(U,O_i)}}^{x_{(O_i,G)}}$ を計算し，それらを $U$ に送る．

*Step 3.* $U$ は，$r_1, r_2, r_{(U,O_i)} \in_R \{0,1\}^{2\ell_n}$ を選択し，$\tilde{t}_1 := t_1 {g_G}^{r_1} {y_{(O_i,G)}}^{r_2}$，$\tilde{t}_2 := (t_2 {P_{(U,O_i)}}^{r_1} {Q_{(U,O_i)}}^{r_2})^{r_{(U,O_i)}}$，$\tilde{P} := {P_{(U,O_i)}}^{r_{(U,O_i)}}$ と $\tilde{Q} := {Q_{(U,O_i)}}^{r_{(U,O_i)}}$ を計算する．次に，$\tilde{e} := \mathcal{H}(g_G, y_{(O_i,G)}, \tilde{P}, \tilde{Q}, \tilde{t}_1, \tilde{t}_2), e := \tilde{e} - r_2$ を計算し，$e$ を $O_i$ に送る．

*Step 4.* $O_i$ は秘密鍵 $x_{(O_i,G)}$ を用いて $s := r - e x_{(O_i,G)}$ を計算し，$U$ に送る．

*Step 5.* $U$ は $t_1 = {g_G}^s {y_{(O_i,G)}}^e$ かつ $t_2 = {P_{(U,O_i)}}^s {Q_{(U,O_i)}}^e$ がなりたつかどうかを検証し，正しければ $\tilde{s} := s + r_1$ を計算し，$P_{(U,O_i)}$ に対する登録証明書が発行されるまで $\sigma_{(U,O_i)} := (\tilde{e}, \tilde{s}, \tilde{P}, \tilde{Q})$ と $r_{(U,O_i)}$ を保管する．

**機関登録証明書発行プロトコル** $CI_O$

$CI_O$ は機関管理者 $M_G \in \boldsymbol{G}$ がユーザ $U$ に機関 $O_i \in \boldsymbol{G}$ の登録証明書，即ち $O_i \in \boldsymbol{G}$ に登録された仮名に対する証明書を発行するのに用いられる．$CI_O$ は，

$$CI_O\langle U(x_U, s_{(U,G)}, s_{(U,O_i)}, P_{(U,O_i)}, r_{(U,O_i)}), M_G(\mathcal{X}_G)\rangle(\mathcal{Y}_G, \mathcal{Y}_{(O_i,G)}, P_{(U,G)}, \sigma_{(U,O_i)})$$
$$\ni [U(\mathcal{C}_{(U,O_i)})]$$

であらわされる．入力 $\sigma_{(U,O_i)} \in GG\langle U, O_i\rangle(P_{(U,O_i)})$ に対し，$P_{(U,O_i)}$ に対する機関登録証明書 $\mathcal{C}_{(U,O_i)}$ をプライベート出力として $U$ に与える．出力される $\mathcal{C}_{(U,O_i)}$ は $C_{(U,O_i)} \equiv (P_{(U,O_i)} {d_G}^{id_{(O_i,G)}} f_G)^{1/E_{(U,O_i)}}$ をみたす $(E_{(U,O_i)}, C_{(U,O_i)})$ の組で与えられる．

ユーザ $U$ はこの時点で，既にグループ $\boldsymbol{G}$ の機関管理者 $M_G$ に仮名 $P_{(U,G)}$，機関 $O_i \in \boldsymbol{G}$ に仮名 $P_{(U,O_i)}$ を登録している．よって，仮名の関連付けを防ぐため，ユーザは機関管理者に $P_{(U,O_i)}$ を見せることなく，$\sigma_{(U,O_i)}$ が $O_i$ によって $U$ の仮名に対して生成されていることを証明する．また，$\sigma_{(U,O_i)}$ は $U$ へのプライベート出力であるため，機関管理者と各機関の結託によるユーザ情報の連結を防ぐことができる．但し，機関管理者は $P_{(U,G)}$ がどの機関に登録しているかという情報のみを管理できる．

プロトコル $CI_O$ は以下のように構成される:

*Step 1.* ユーザ $U$ はランダムに選択した素数 $E_{(U,O_i)} \in_R \Lambda$ と $r \in_R \mathbb{Z}_{n_G}, id_{(O_i,G)} \in \Delta$ に対し，$c := r^{E_{(U,O_i)}} P_{(U,O_i)} {d_G}^{id_{(O_i,G)}} f_G$ を計算し，$c, E_{(U,O_i)}, \sigma_{(U,O_i)}$ を $G$ に送

る．さらに，その $\sigma_{(U,O_i)}$ に対応するユーザであることを証明する: $U$ はランダムに選択した $r_1 \in_R \{0,1\}^{2\ell_n}$ に対して $c_1 := re_G{}^{r_1}$ を計算し，$\boldsymbol{G}$ に登録されている仮名 $P_{(U,G)}$ に対して以下の $PK^2$ を実行する:

$$\begin{aligned} PK^2\{(\alpha,\beta,\gamma,\delta,\varepsilon,\zeta,\xi,\eta) \quad : \quad & P_{(U,G)} = g_G{}^{\alpha} h_G{}^{\beta} \\ \wedge \quad & 1 = P_{(U,G)}{}^{\gamma} / g_G{}^{\delta} h_G{}^{\varepsilon} \\ \wedge \quad & \tilde{P} = g_G{}^{\delta} h_G{}^{\zeta} \\ \wedge \quad & \tilde{P} = c^{\gamma} (e_G{}^{E_{(U,O_i)}})^{\xi} / (c_1{}^{E_{(U,O_i)}} d_G{}^{id_{(O_i,G)}} f_G)^{\gamma} \\ \wedge \quad & \alpha \in \Gamma, \beta \in \Delta\}. \end{aligned}$$

*Step 2.* $M_G$ は $t_1' := g_G{}^{\tilde{s}} y_{(O_i,G)}{}^{\tilde{e}}, t_2' := \tilde{P}^{\tilde{s}} \tilde{Q}^{\tilde{e}}$ に対して $\tilde{e} = \mathcal{H}(g_G, y_{(O_i,G)}, \tilde{P}, \tilde{Q}, t_1', t_2')$ が成り立つかどうか検証し，正しければ $c' := c^{1/E_{(U,O_i)}}$ とし，$U$ に送る．

*Step 3.* $U$ は $C_{(U,O_i)} := c'/r$ を計算し，$C_{(U,O_i)}{}^{E_{(U,O_i)}} \equiv P_{(U,O_i)} d_G{}^{id_{(O_i,G)}} f_G \pmod{n_G}$ が成り立つかどうか検証する．正しければ $(E_{(U,O_i)}, C_{(U,O_i)})$ を機関 $O_i$ の登録証明書として保管する．

### 機関登録証明プロトコル $CS^+$

$CS^+$ は，ユーザ $U$ が検証者 $V$ に機関 $O_i \in \boldsymbol{G}$ への登録を証明するのに用いられる．

$$CS^+\langle U(x_U, s_{(U,O_i)}, P_{(U,O_i)}, \mathcal{C}_{(U,O_i)}), V\rangle(\mathcal{Y}_G, y_{(O_i,G)}) \ni \{1,0\}$$

は，$P_{(U,O_i)} \in PG\langle U, O_i\rangle$ かつ $\mathcal{C}_{(U,O_i)} \in CI_O\langle U(P_{(U,O_i)}), M_G\rangle$ をみたす $P_{(U,O_i)}$ と $\mathcal{C}_{(U,O_i)}$ の入力に対して圧倒的確率で 1 を出力する．

プロトコル $CS^+$ は以下のように構築される:

*Step 1.* $U$ は $r_1, r_2 \in_R \{0,1\}^{2\ell_n}$ に対し，$c_1 := C_{(U,O_i)} e_G{}^{r_1}$ と $c_2 := e_G{}^{r_1} d_G{}^{r_2}$ を計算し，それらを $V$ におくる．

*Step 2.* $U$ は以下の $PK^2$ によって，検証者 $V$ に機関 $O_i$ の登録証明をおこなう:

$$\begin{aligned} PK^2\{(\alpha,\beta,\gamma,\delta,\varepsilon,\zeta,\xi) \quad : \quad & f_G d_G{}^{id_{(O_i,G)}} = c_1{}^{\alpha} / g_G{}^{\beta} h_G{}^{\gamma} e_G{}^{\delta} \\ \wedge \quad & c_2 = e_G{}^{\varepsilon} d_G{}^{\zeta} \\ \wedge \quad & 1 = c_2{}^{\alpha} / e_G{}^{\delta} d_G{}^{\xi} \\ \wedge \quad & \alpha \in \Lambda \wedge \beta \in \Gamma \wedge \gamma \in \Delta\}. \end{aligned}$$

**機関情報をもたない機関登録証明プロトコル** $CS^-$

$CS^-$ は，ユーザ $U$ が検証者 $V$ にグループ $\boldsymbol{G}$ 内のある機関に登録していることを証明するのに用いられる．

$$CS^-\langle U(x_U, s_{(U,O_i)}, P_{(U,O_i)}, \mathcal{C}_{(U,O_i)}, \mathcal{Y}_{(O_i,G)}), V\rangle(\mathcal{Y}_G) \ni \{1,0\}$$

は，$P_{(U,O_i)} \in PG\langle U, O_i\rangle$ かつ $\mathcal{C}_{(U,O_i)} \in CI_O\langle U(P_{(U,O_i)}), M_G\rangle$ をみたす $P_{(U,O_i)}$ と $\mathcal{C}_{(U,O_i)}$ の入力に対して圧倒的確率で 1 を出力する．

プロトコル $CS^-$ は以下のように構築される:

*Step 1.* $U$ は，$r_1, r_2 \in_R \{0,1\}^{2\ell_n}$ に対し，$c_1 := C_{(U,O_i)}{e_G}^{r_1}$ と $c_2 := {e_G}^{r_1}{d_G}^{r_2}$ を計算し，それらを $V$ に送る．

*Step 2.* $U$ は以下の $PK^2$ によって，検証者 $V$ に機関情報をもたない機関登録証明をおこなう:

$$\begin{aligned} PK^2\{(\alpha,\beta,\gamma,\delta,\varepsilon,\zeta,\xi,\eta) \quad : \quad & f_G = {c_1}^{\alpha}/{g_G}^{\beta}{h_G}^{\gamma}{d_G}^{\delta}{e_G}^{\varepsilon} \\ \wedge \quad & c_2 = {e_G}^{\zeta}{d_G}^{\xi} \\ \wedge \quad & 1 = {c_2}^{\alpha}/{e_G}^{\varepsilon}{d_G}^{\eta} \\ \wedge \quad & \alpha \in \Lambda \wedge \beta \in \Gamma \wedge \gamma \in \Delta\}. \end{aligned}$$

**検証機関への機関登録認証プロトコル** $CT^+$

$CT^+$ において，ユーザ $U$ はエンティティ $X \in \{O_j \in \boldsymbol{G}_J, M_{G_J} \in \boldsymbol{G}_J\}$ に機関 $O_i \in \boldsymbol{G}_I$ への登録を示すとともに，$X$ における $P_{(U,X)}$ に対応するユーザであることを示すことができる．

$$CT^+\langle U(x_U, s_{(U,O_i)}, s_{(U,O_j)}, P_{(U,O_i)}, \mathcal{C}_{(U,O_i)}), X\rangle(\mathcal{Y}_{G_I}, \mathcal{Y}_{G_J}, y_{(O_i,G_I)}, P_{(U,X)}) \ni \{1,0\}$$

は，$\mathcal{C}_{(U,O_i)} \in CI_O\langle U(P_{(U,O_i)}), M_{G_J}\rangle$, $P_{(U,O_i)} \in PG\langle U(x_U), O_i\rangle$ かつ $P_{(U,X)} \in PG\langle U(x_U), X\rangle$ をみたす $P_{(U,O_i)}, P_{(U,X)}$ と $\mathcal{C}_{(U,O_i)}$ に対して圧倒的確率で 1 を出力する．

プロトコル $CT^+$ は以下のように構築される:

*Step 1.* $U$ は，$r_1, r_2 \in_R \{0,1\}^{2\ell_n}$ に対して，$c_1 := C_{(U,O_i)}{e_{G_I}}^{r_1}$ と $c_2 := {e_{G_I}}^{r_1}{d_{G_I}}^{r_2}$ を計算し，それらを $X$ に送る．

*Step 2.* $U$ は，以下の $PK^2$ によって機関登録証明をおこなう:

$$\begin{aligned} PK^2\{(\alpha,\beta,\gamma,\delta,\varepsilon,\zeta,\xi,\eta) \quad : \quad & f_{G_I} {d_{G_I}}^{id_{(O_i,G_I)}} = {c_1}^{\alpha} / {g_{G_I}}^{\beta} {h_{G_I}}^{\gamma} {e_{G_I}}^{\delta} \\ \wedge \quad & c_2 = {e_{G_I}}^{\varepsilon} {d_{G_I}}^{\zeta} \\ \wedge \quad & 1 = {c_2}^{\alpha} / {e_{G_I}}^{\delta} {d_{G_I}}^{\xi} \\ \wedge \quad & P_{(U,X)} = {g_{G_J}}^{\beta} {h_{G_J}}^{\eta} \\ \wedge \quad & \alpha \in \Lambda \wedge \beta \in \Gamma \wedge \gamma \in \Delta\}. \end{aligned}$$

**検証機関への機関登録証明プロトコル** $CT^-$

$CT^-$ において，ユーザ $U$ はエンティティ $X \in \{O_j \in \boldsymbol{G}_J, M_{G_J} \in \boldsymbol{G}_J\}$ にグループ $\boldsymbol{G}_I$ 内のある機関への登録を示すとともに，$X$ における $P_{(U,X)}$ における $P_{(U,X)}$ に対応するユーザであることを示すことができる．

$$CT^-\langle U(x_U, s_{(U,O_i)}, s_{(U,O_j)}, P_{(U,O_i)}, \mathcal{C}_{(U,O_i)}, y_{(O_i,G_I)}), X\rangle(\mathcal{Y}_{G_I}, \mathcal{Y}_{G_J}, P_{(U,X)}) \ni \{1,0\}$$

は，$\mathcal{C}_{(U,O_i)} \in CI_O\langle U(P_{(U,O_i)}), M_{G_I}\rangle$, $P_{(U,O_i)} \in PG\langle U(x_U), O_i\rangle$ かつ $P_{(U,X)} \in PG\langle U(x_U), O_j\rangle$ をみたす $P_{(U,O_i)}, P_{(U,X)}$ と $\mathcal{C}_{(U,O_i)}$ に対して圧倒的確率で 1 を出力する．

プロトコル $CT^-$ は以下のように構築される:

**1.** $U$ は $r_1, r_2 \in_R \{0,1\}^{2\ell_n}$ に対し，$c_1 := C_{(U,O_i)} {e_{G_I}}^{r_1}, c_2 := {e_{G_I}}^{r_1} {d_{G_I}}^{r_2}$ を計算し，$c_1$ と $c_2$ を $X$ に送信する．

**2.** $U$ は，以下の $PK^2$ によって機関登録証明をおこなう:

$$\begin{aligned} PK^2\{(\alpha,\beta,\gamma,\delta,\varepsilon,\zeta,\xi,\eta,\varphi) \quad : \quad & f_{G_I} = {c_1}^{\alpha} / {g_{G_I}}^{\beta} {h_{G_I}}^{\gamma} {d_{G_I}}^{\delta} {e_{G_I}}^{\varepsilon} \\ \wedge \quad & c_2 = {e_{G_I}}^{\zeta} {d_{G_I}}^{\xi} \\ \wedge \quad & 1 = {c_2}^{\alpha} / {e_{G_I}}^{\varepsilon} {d_{G_I}}^{\eta} \\ \wedge \quad & P_{(U,X)} = {g_{G_J}}^{\beta} {h_{G_J}}^{\varphi} \\ \wedge \quad & \alpha \in \Lambda \wedge \beta \in \Gamma \wedge \gamma \in \Delta\}. \end{aligned}$$

### 5.3.3 安全性の考察

本章では，提案するシステムの安全性について考察する．強 RSA 問題仮定と Diffie-Hellman 決定問題仮定の下，以下の補題が導かれる:

**補題 1** $PG$ における

$$\begin{aligned} PK^2\{(\alpha,\beta,\gamma,\delta,\varepsilon,\zeta,\vartheta,\xi) \quad : \quad & c_1 = d^{\alpha}e^{\beta} \\ \wedge \quad & c_2 = d^{\gamma}e^{\delta} \\ \wedge \quad & c_3 = d^{\varepsilon}e^{\zeta} \\ \wedge \quad & P_{(U,O_i)} = g^{\gamma}h^{\vartheta} \\ \wedge \quad & (c_1 d^{r-2^{\ell_\Delta+1}})/({c_3}^{2^{\ell_\Delta+1}+1}) = d^{\vartheta}e^{\xi} \\ \wedge \quad & \gamma \in \Gamma \wedge \vartheta \in \Delta\} \end{aligned}$$

は $P_{(U,O_i)}$ を生成する正しい $x_U, s_{(U,O)}$ の統計的ゼロ知識証明である．

*Proof.* $PK^2$ プロトコルの性質より，述語部分をみたす $\alpha,\beta,\gamma,\delta,\varepsilon,\zeta,\vartheta,\xi$ を抽出することができる．また，$c_1 = d^{\alpha}e^{\beta}, c_3 = d^{\varepsilon}e^{\zeta}, (c_1 d^{r-2^{\ell_\Delta+1}})/({c_3}^{2^{\ell_\Delta+1}+1}) = d^{\vartheta}e^{\xi}$ より，$d^{\alpha+r-2^{\ell_\Delta+1}}e^{\beta}/(d^{\varepsilon(2^{\ell_\Delta+1}+1)}e^{\zeta(2^{\ell_\Delta+1}+1)}) = d^{\alpha+r-2^{\ell_\Delta}+1-\varepsilon(2^{\ell_\Delta+1}+1)} \cdot e^{\beta-\zeta(2^{\ell_\Delta+1}+1)} = d^{\vartheta}e^{\xi}$．よって，$\vartheta \in \Delta$ かつ $\alpha + r - 2^{\ell_\Delta+1} + 1 - \varepsilon(2^{\ell_\Delta+1}+1) = \vartheta$ より $\vartheta = (\alpha + r \bmod 2^{\ell_\Delta+1}+1) - 2^{\ell_\Delta} + 1$ を得る．

**補題 2** $CS_G, CT_G, CS^+, CS^-, CT^+$ と $CT^-$ における $PK^2$ プロトコルは，証明者の正しい型の秘密鍵，秘密情報，及び属性証明書の統計的ゼロ知識証明である．

これらの証明は，[11] における証明と同様であり，ここでは，$TC^-$ における $PK^2$ プロトコルに関する補題及びその証明を与えることにする．

**補題 3** $TC^-$ における

$$\begin{aligned} PK^2\{(\alpha,\beta,\gamma,\delta,\varepsilon,\zeta,\xi,\eta,\varphi) \quad : \quad & {f_{G_I}} = {c_1}^{\alpha}/{g_{G_I}}^{\beta}{h_{G_I}}^{\gamma}{d_{G_I}}^{\delta}{e_{G_I}}^{\varepsilon} \\ \wedge \quad & c_2 = {e_{G_I}}^{\zeta}{d_{G_I}}^{\xi} \\ \wedge \quad & 1 = {c_2}^{\alpha}/{e_{G_I}}^{\varepsilon}{d_{G_I}}^{\eta} \\ \wedge \quad & P_{(U,X)} = {g_{G_J}}^{\beta}{h_{G_J}}^{\varphi} \\ \wedge \quad & \alpha \in \Lambda \wedge \beta \in \Gamma \wedge \gamma \in \Delta\} \end{aligned}$$

は，$P_{(U,X)} = {g_{G_J}}^{x}{h_{G_J}}^{s_1} \pmod{n_{G_J}}$ と $C^E = {g_{G_I}}^{x}{h_{G_I}}^{s_2}{d_{G_I}}^{y}f_{G_I} \pmod{n_{G_I}}$ をみたす $x \in \Gamma, s_1, s_2 \in \Delta, E \in \Lambda, C, y$ の統計的ゼロ知識証明である．

*Proof.* $PK^2$ プロトコルの性質より，述語部分をみたす $\alpha,\beta,\gamma,\delta,\varepsilon,\zeta,\xi,\eta,\varphi$ を抽出することができる．また，$c_2 = {e_{G_I}}^{\zeta}{d_{G_I}}^{\xi}, 1 = {c_2}^{\alpha}/{e_{G_I}}^{\varepsilon}{d_{G_I}}^{\eta}$ より $\zeta\alpha = \varepsilon \pmod{ord(e_{G_I})}$

を得る．よって，${c_1}^{\alpha}/{e_{G_I}}^{\varepsilon} = {g_{G_I}}^{\beta}{h_{G_I}}^{\gamma}{d_{G_I}}^{\delta}f_G = (c_1/{e_{G_I}}^{\zeta})^{\alpha}$ かつ $\alpha \in \Lambda, \beta \in \Gamma, \gamma \in \Delta$ より，証明者は正しい属性証明書 $c_1/{e_{G_I}}^{\zeta}$ を保持しており，加えて $P_{(U,X)} = {g_{G_J}}^{\beta}{h_{G_J}}^{\eta}$ より，その証明書に対応する仮名と $P_{(U,X)}$ は同じ鍵を用いて生成されていることがいえる．

**補題 4** $CI_O$ における

$$\begin{aligned} PK^2\{(\alpha,\beta,\gamma,\delta,\varepsilon,\zeta,\xi,\eta) \quad : \quad & P_{(U,G)} = {g_G}^{\alpha}{h_G}^{\beta} \\ \wedge \quad & 1 = {P_{(U,G)}}^{\gamma}/{g_G}^{\delta}{h_G}^{\varepsilon} \\ \wedge \quad & \tilde{P} = {g_G}^{\delta}{h_G}^{\zeta} \\ \wedge \quad & \tilde{P} = c^{\gamma}({e_G}^{E_{(U,O_i)}})^{\xi}/({c_1}^{E_{(U,O_i)}}{d_G}^{id_{(U,O_i)}}f_G)^{\gamma} \\ \wedge \quad & \alpha \in \Gamma, \beta \in \Delta\} \end{aligned}$$

は，$P_{(U,G)} = {g_G}^{x}{h_G}^{s}$, $c = r^{E_{(U,O_i)}}P_{(U,O_i)}{d_G}^{id_{(O_i,G)}}f_G$ かつ $\tilde{P} = {P_{(U,O_i)}}^{r_{(U,O_i)}}$ をみたす $x \in \Gamma, s \in \Delta, r, r_{(U,O_i)}$ の統計的ゼロ知識証明である．

*Proof.* $P_{(U,G)} = {g_G}^{\alpha}{h_G}^{\beta}$, $1 = {P_{(U,G)}}^{\gamma}/{g_G}^{\delta}{h_G}^{\varepsilon}$ から $\alpha\gamma \equiv \delta \pmod{ord(g_G)}$ を得ることができ，$\tilde{P} = c^{\gamma}({e_G}^{E_{(U,O_i)}})^{\xi}/({c_1}^{E_{(U,O_i)}}{d_G}^{id_{(O_i,G)}}f_G)^{\gamma}$ より，$c^{\gamma} = {g_G}^{\delta}{h_G}^{\zeta}({c_1}^{E_{(U,O_i)}}{d_G}^{id_{(O_i,G)}}f_G)^{\gamma}/({e_G}^{E_{(U,O_i)}})^{\xi} = \{(c_1/{e_G}^{\xi/\gamma})^{E_{(U,O_i)}}{g_G}^{\alpha}{h_G}^{\zeta/\gamma}{d_G}^{id_{(O_i,G)}}f_G\}^{\gamma}$ を導く．これによって，$c$ が $P_{(U,G)}$ を生成する秘密情報 $\alpha$ を用いたある仮名 ${g_G}^{\alpha}{h_G}^{\zeta/\gamma}$ と $id_{(O_i,G)}$ を用いて正しい型で生成されているといえる．

### 5.3.4 シミュレータの構築

提案する匿名証明書方式の安全性を示すためにシミュレータを構築し，それが安全なシステムの定義を満たすことを示す必要がある．ここで，攻撃者 $\mathcal{A}$ がコントロールするエンティティは一つのエンティティに包摂されるものとし，$\mathcal{A}$ のためのシミュレータの構築をおこなう．

**初期設定**

$\mathcal{A}$ にコントロールされない機関管理者 $M_G \in \boldsymbol{G}$ と機関 $O \in \boldsymbol{G}$ において，$\mathcal{S}$ はそれらの秘密鍵と公開鍵の組 $(\mathcal{X}_G, \mathcal{Y}_G)$ と $(x_{(O_i,G)}, y_{(O_i,G)})$ を用意する．さらに，$\mathcal{S}$ は $\mathcal{A}$ によってコントロールされたユーザの登録を記録する $archive_G$ と $archive_O$ を作成する．$archive_G$ はグループ $\boldsymbol{G}$ の登録証明書，$archive_O$ は

機関 $O$ の登録証明書の発行をうけたユーザのデータをそれぞれ記録するものとする．さらに，$\mathcal{A}$ によってコントロールされるユーザのリスト $list_{\mathcal{A}}$ を初期化しておく．

**グループへの仮名登録**

ユーザは機関管理者 $M_G$ に登録する仮名を生成する:

(I) $\mathcal{A}$ によってコントロールされるユーザが正直な機関管理者に仮名を登録する場合，(i) $\mathcal{S}$ は，プロトコル $PG$ における知識抽出によって，ユーザの秘密鍵 $x$ と秘密情報 $s$ を得る．(i-1) その秘密鍵 $x$ に対応するユーザが $list_{\mathcal{A}}$ に存在しなければ，$\mathcal{S}$ はログインネームを $L_U$ とする新しいユーザを作成し，$T$ と対話することによって仮名 $N_{(U,G)}$ と，$L_U$ に対応する鍵 $K_U$ を手に入れる．さらに，$(x := x_U, s := s_{(U,G)})$ とし，$\mathcal{S}$ は $\mathcal{A}$ にコントロールされたユーザのリスト $list_{\mathcal{A}}$ に $(U, L_U, K_U, N_{(U,G)}, x_U, s_{(U,G)})$ を加える．(ii-2) $x$ に対応するユーザが $list_{\mathcal{A}}$ に既に存在するならば，$\mathcal{S}$ は，$T$ と対話することによって $x$ に対応するユーザ $U$ のために仮名 $N_{(U,G)}$ を手に入れ，$U$ の情報に $N_{(U,G)}, s_{(U,G)} := s$ を加える．

(II) $T$ を通して正直なユーザが $\mathcal{A}$ によってコントロールされたグループ管理者に登録する仮名を作成する場合，$\mathcal{S}$ はプロトコル $PG$ におけるゼロ知識シミュレータを用いて，$\mathcal{A}$ の動きをシミュレートする．

**グループ登録証明書の発行**

ユーザは機関管理者 $M_G$ に証明書の発行を要求する:

(I) $\mathcal{A}$ にコントロールされたユーザが正直なグループ管理者に証明書の発行を要求する場合，(i) $T$ からのメッセージを受け取るとすぐに，$\mathcal{S}$ は $x$ と $s$ の値を得るためにプロトコル $CI_G$ の *Step 1* における知識抽出をおこなう．$(x, s)$ に対応する仮名 $N$ に対して，(i-1) $N \notin list_{\mathcal{A}}$ ならば， $\mathcal{S}$ は証明書の発行を拒否する．(i-2) $N \in list_{\mathcal{A}}$ ならば，$\mathcal{S}$ は $T$ と対話することによって正当な証明書 $(E, C)$ を発行する．$\mathcal{S}$ は $archive_G$ 内の $N$ に対応するデータに $(E_{(U,G)}, C_{(U,G)}) := (E, C)$ を追加する．

(II) 正直なユーザが $T$ を通して $\mathcal{A}$ にコントロールされた機関管理者 $M_G$ に証明書の発行を要求する場合，$\mathcal{S}$ はプロトコル $CI_G$ の *Step 1* におけるゼロ

知識シミュレータを走らせ，$U$ がおこなうようにプロトコルを続ける．もし，$U$ が受理すれば $\mathcal{S}$ は $T$ に証明書が発行されたことを知らせる．

**機関への仮名登録**

ユーザは機関 $O \in \boldsymbol{G}$ に登録する仮名を生成する:

このシミュレーションは上のグループへの仮名登録と同様におこなうことができる．結果として $archive_O$ に $(U, L_U, K_U, N_{(U,O)}, x_U, s_{(U,O)})$ が保管される．

**仮名保証書の生成**

ユーザは仮名の保証書の生成を機関 $O$ に要求する:

(I) ユーザが $\mathcal{A}$ によってコントロールされている場合，(i) $\mathcal{S}$ はユーザの鍵 $x$ と秘密情報 $s$ を得るためにプロトコル $GG$ の *Step 1* の知識抽出をおこなう．

$(x, s)$ に対応する $N$ に対して，(i-1) $N \notin list_{\mathcal{A}}$ ならば，$\mathcal{S}$ は保証書の発行を拒否する．(i-2) $N \in list_{\mathcal{A}}$ ならば，$\mathcal{S}$ は $T$ との対話によって $\sigma$ を作成し，$archive_O$ 内の $N$ に対応するデータに $\sigma_{(U,O)} := \sigma$ を加える．

(II) 機関 $O$ が $\mathcal{A}$ によってコントロールされている場合，$\mathcal{S}$ はプロトコル $GG$ の *Step 1* におけるゼロ知識シミュレータを走らせプロトコルにおける $U$ の動きをシュミレートする．

**機関属性証明書の発行**

ユーザは $O \in \boldsymbol{G}$ のへの登録証明書発行を機関管理者 $M_G \in \boldsymbol{G}$ に要求する:

(I) ユーザが $\mathcal{A}$ にコントロールされている場合，(i) $T$ からのメッセージを受け取るとすぐに，$\mathcal{S}$ は $x, s_{(U,G)}$ と $r$ を手に入れるために プロトコル $CI_G$ の *Step 1* における知識抽出をおこなう．(i-1) $(x, s_{(U,G)}) \notin archive_G$ ならば，$\mathcal{S}$ は証明書の発行を拒否する．(i-2) $(x, s_{(U,G)}) \in archive_G$ ならば，$\mathcal{S}$ は $G$ の残りの動きをおこない，$E$ に対応する $c'$ を発行し，$c'/r$ によって $C$ を求める．$\mathcal{S}$ は $archive_O$ 内の $(x, s_{(U,O)})$ に対応するデータに $(C_{(U,O)}, E_{(U,O)}) := (C, E)$ を加える．

(II) もしユーザが $\mathcal{A}$ によってコントロールされており，機関 $O \in \boldsymbol{G}$ が不正直である場合，(i) $T$ からのメッセージを受け取るとすぐに，$\mathcal{S}$ はプロトコル $CI_G$ の *Step 1* での知識抽出をおこない $x, s_{(U,G)}$ と $r$ を得，$archive_O$ をチェッ

クする．(i-1) もし $(x, s) \in archive_O$ ならば，$\mathcal{S}$ はこのユーザを $U$ とする．(i-2) もし $(x, s) \notin archive_O$ ならば，$U$ を $x$ に対応するユーザとし，$T$ と対話することによって $N_{(U,O)}$ を得る．(ii) $\mathcal{S}$ は $archive_G$ をチェックする．(ii-1) もし $(x, s_{(U,G)}) \notin archive_G$ ならば，$\mathcal{S}$ は証明書の発行を拒否する．(ii-2) もし $(x, s_{(U,G)}) \in archive_G$ ならば $\mathcal{S}$ は $G$ の残りの作業をおこない，$E$ に対応する正しい $c'$ を作成する．$\mathcal{S}$ は $c'/r$ によって $C$ を求め，$(x, s_{(U,O)}, C, E)$ を $archive_O$ 内の $x$ に対応するデータに $(C_{(U,O)}, E_{(U,O)}) := (C, E)$ を加える．

(III) もし，機関管理者 $M_G \in \boldsymbol{G}$ が $\mathcal{A}$ によってコントロールされている場合，$\mathcal{S}$ は $CI_O$ の *Step 1* をゼロ知識シミュレートし，ユーザ側の残りの行動をおこなう．もし，ユーザが受理するならば，$\mathcal{S}$ は $T$ に証明書が発行されたことを知らせる．

**機関登録証明，機関情報をもたない機関登録証明，検証機関に対する機関登録証明**

これらのシミュレーションは，以下の検証機関に対する機関情報をもたない機関登録証明におけるシミュレーションから容易に導くことができる．

**検証機関に対する機関情報をもたない機関登録証明**

ユーザは検証機関 $O_j \in \boldsymbol{G}_J$ に $\boldsymbol{G}_I$ 内のある機関の登録証明をおこなう:

(I) ユーザが $\mathcal{A}$ によってコントロールされており，登録証明書を発行した $M_{G_I} \in \boldsymbol{G}_I$ は正直であり，検証機関 $O_j$ も正直である場合，(i) $\mathcal{S}$ は $CT^-$ の $O_j$ 側の動きをおこない，知識抽出によって $x, s_{(U,O_i)}, s_{(U,O_j)}, E, C$ と $y_{(O_i,G_I)}$ を得る．(i-1) $(x, s_{(U,O_i)}, E, C) \notin archive_{O_i}$ ならば $\mathcal{S}$ は拒否する．(i-2) $(x, s_{(U,O_i)}, E, C) \in archive_{O_i}$ ならば $\mathcal{S}$ は $U$ による保証書の保持証明を $T$ を通じておこなう．

(II) ユーザは $\mathcal{A}$ にコントロールされており，登録証明書を発行した $M_{G_I} \in \boldsymbol{G}_I$ は不正直であり，検証機関 $O_j$ が正直である場合，(i) $\mathcal{S}$ は $x, s, s_{(U,O_j)}, E, C, y$ を手に入れるため，知識抽出を用いて $CT^-$ における $O_j$ の動きをシュミレートする．ここで，$y$ を公開鍵としてもつ機関を $O_i$ とする．(i-1) もし $O_j$ 側がプロトコルの実行を拒否した場合，$\mathcal{S}$ は何もしない．(i-2) そうでない場合: (2-A-a) もし $x \in archive_{O_i}$ ならば，このユーザを $U$ とする．(2-A-b) もし $x \notin archive_{O_i}$ ならば，$U$ を $x$ に対応するユーザとし $\mathcal{S}$ は $T$ との対話によって

仮名 $N_{(U,O_i)}$ を手に入れる．(2-B) もし $(E,C) \notin archive_{O_i}$ ならば，$\mathcal{S}$ は $CI_O$ を走らせ $U$ の記録にその出力を加える．(2-C) $\mathcal{S}$ は $U$ による証明書の保持証明において $T$ と通信する．

(III) もし検証機関 $O_j$ が $\mathcal{A}$ によってコントロールされている場合，$\mathcal{S}$ はゼロ知識シミュレータを走らせる．

**定理 5** 強 RSA 問題仮定のもと，正しい仮名と証明書の組 $(P,C,E)$ の保持証明をおこなうことができるならば，この組はプロトコル $PG\langle U,O_i\rangle, CI_O\langle U,M_G\rangle$ によって生成されたものである．

*Proof.* $K_A$ を攻撃者が要求できる証明書の数，$K_H$ を正直なユーザが要求できる証明書の数とし，$K = K_A + K_H$ とおく．

$\mathcal{A}$ を適応的に $O_i$ とプロトコル $PG$，また $M_G$ とプロトコル $CI_G$ を動かし，$K$ 個の仮名と証明書の組 $(P_j,(C_j,E_j))$ $(1 \leq j \leq K)$ を得ることができる攻撃者であるとする．このとき，$\mathcal{A}$ がある正直な検証者 $V$ または正直な検証機関 $O_j$ に $\tilde{x} \in \Gamma, \tilde{s} \in \Delta, \tilde{E} \in \Lambda$ であり，かつ $1 \leq j \leq K$ に対して $(g^{\tilde{x}}h^{\tilde{s}}, \tilde{E}) \neq (P_j, E_j)$ をみたす $(\tilde{x},\tilde{s},\tilde{C},\tilde{E})$ の保持証明をおこなうことができると仮定する．ここで，以下の 2 つの場合について議論を進める:

**1.** $\forall j$ に対して $\gcd(E_j,\tilde{E}) = 1$ のとき: もし $\mathcal{A}$ がこれらの保持証明に成功するならば，Game 1 を用いて強 RSA 問題を解くことができる．

**2.** $\gcd(E_j,\tilde{E}) = E_j$ を満たす $j$ が存在するとき: $\mathcal{A}$ がこれらの保持証明に成功するならば，Game 2 を用いて強 RSA 問題を解くことができる．

以下に Game 1 と Game 2 を述べる．

**Game 1:** RSA の法 $n$ と $z \in \mathbb{Z}_n^{*2}$ を用意する．ここで，$n$ は素数 $p', q'$ に対する素数 $p = 2p'+1$ と $q = 2q'+1$ の積であり $z$ を位数 $p'q'$ をもつ元とする．

**1.** $v_1,\ldots,v_K \in_R ]-2^{\ell_\Delta}+2^{2\ell_\Gamma}, 2^{\ell_\Delta}[$ と素数 $E_1,\ldots,E_K \in_R \Delta$ を選択する．

**2.** $h := z^{\prod_\ell E_\ell} \bmod n$ とする．

**3.** $r_1,r_2,r_3 \in_R \Gamma$ を選択し，$g := h^{r_1} \bmod n, d := h^{r_2} \bmod n, f := h^{r_3} \bmod n$ とする．

**4.** 全ての $1 \leq i \leq K$ に対し，$C_i := z^{(v_i+r_3)\prod_{\ell \neq i} E_\ell}$ を計算する．

**5.** $e \in_R \mathbb{Z}_n^{*2}$ を選択し，$(n, d, e, f, g, h)$ をグループの公開鍵とする．

**6.** 正直なユーザが $M_G$ に証明書の発行を求めてきた場合は，$(P_j = {C_j}^{E_j}/f, E_j, C_j)$ を選ぶ．

**7.** $\mathcal{A}$ と以下のように対話をおこなう:

仮名生成プロトコル $PG$: *Step 1*では$\mathcal{A}$からコミットメント $c_1, c_2$ を受け取り，$PK^2$ から ${c_1}^2 = (d^2)^{\tilde{r}_j}(e^2)^{\tilde{r}'_j}$ かつ ${c_2}^2 = (d^2)^{\tilde{x}_j}(e^2)^{\tilde{r}''_j}$ をみたす $\tilde{x}_j, \tilde{r}_j, \tilde{r}'_j, \tilde{r}''_j$ を抽出する．*Step 2* において，$s_j = v_j - \tilde{x}_j r_1 \in \mathbb{Z}$ を計算する．このとき $s_j \in \Delta$ である．次に，$r = s_j + (2^{\ell_\Delta} - 1) - \tilde{r}_j \bmod (2^{\ell_\Delta+1} - 1)$ を計算し，$r$ を $\mathcal{A}$ に送る．残りはプロトコルに沿っておこなう．

証明書発行プロトコル $CI_O$: $\mathcal{A}$から仮名 $P$ を受け取ったあと，$P = h^{v_j}$ をみたす $j$ を求める．もしこれをみたす $j$ が存在しない場合は，$CI_O$ の実行を中止する．そうでない場合は $(C_j, E_j)$ を $\mathcal{A}$ に送る．

登録証明プロトコル $CS$ と $CT$: 検証者 $V$ または検証機関 $O_j$ として実行する．それぞれの *Step 2* における $PK^2$ において，

$$({\tilde{c_1}}^2/(e^2)^{\tilde{y}})^{\tilde{E}} \equiv (g^2)^{\tilde{x}}(h^2)^{\tilde{s}}(d^2)^{id} f^2$$

をみたす $\tilde{x} \in \Gamma, \tilde{s} \in \Delta, \tilde{E} \in \Lambda, \tilde{y}$ と $\tilde{c_1}$ の抽出をおこなう．

ここで，$\gcd(\tilde{E}, E_j) \neq 1$ となる $E_j (1 \leq j \leq K)$ が存在するならば，$V$ または $O_\ell$ としてプロトコルの実行を続ける．そうでない場合は $\hat{E} := 2(\tilde{x}r_1 + \tilde{s} + r_2 id + r_3)\prod_\ell E_\ell$ とする．このとき，$\gcd(\tilde{E}, E_j) = 1 (1 \leq j \leq K)$ より，$w := \gcd(\tilde{E}, \hat{E}) = \gcd(\tilde{E}, 2(\tilde{x}r_1 + \tilde{s} + r_2 id + r_3))$ を得る．拡張ユークリッドの互助法を用いて，$\alpha\tilde{E} + \beta\hat{E} = w$ をみたす $\alpha, \beta \in \mathbb{Z}$ を求め，$u := z^\alpha({\tilde{c_1}}^2/(e^2)^{\tilde{y}})^\beta \bmod n$，$E := \tilde{E}/w$ とおく．このとき，$|2(\tilde{x}r_1 + \tilde{s} + r_2 id + r_3))| < 2(2^{2\ell_\Gamma} + 2^{\ell_\Gamma} + 2^{\ell_{2\Delta}} + 2^{\ell_\Delta}) < 2^{\ell_\Lambda}$ かつ $\tilde{E} \in \Lambda$ より $E > 1$ である．また，$({\tilde{c_1}}^2/(e^2)^{\tilde{y}})^{\tilde{E}} \equiv (g^2)^{\tilde{x}}(h^2)^{\tilde{s}}(d^2)^{id} f^2 \equiv (z^2)^{(\tilde{x}r_1 + \tilde{s} + r_2 + r_3 id)\prod_\ell E_\ell}$ であることから，$u^E \equiv (z^\alpha(\tilde{c_1}/e^{\tilde{y}})^\beta)^{\tilde{E}/w} \equiv z^{(alpha\tilde{E} + \beta\hat{E})/w} \equiv z \pmod n$ を得る．この $(u, E)$ は強 RSA 問題の解であり，これらを出力して停止する．

機関情報をもたない登録証明プロトコル $CS^-$ と $CT^-$: これらのプロトコルからも，登録証明プロトコル $CS^+, CT^+$ と同様に $u^E \equiv z \pmod n$ をみたす $(u, E)$ を出力することができる．

**8.** もし $\mathcal{A}$ が停止したならば，Null を返し停止する．

**Game 2:** RSA の法 $n$ と $z \in \mathbb{Z}_n^{*2}$ を Game 1 と同様に設定する．

**1.** $v_1, \ldots, v_K \in_R ]-2^{\ell_\Delta}+2^{2\ell_\Gamma}, 2^{\ell_\Delta}[$ と素数 $E_1, \ldots, E_K \in_R \Gamma$ を選択する．

**2.** $k \in_R \{1, \ldots, K\}$ を選択し，$h := z^{\prod_{\ell \neq k} E_\ell} \bmod n$ とする．

**3.** $r_1, r_2, r_3 \in_R \Gamma$ を選択し，$g := h^{r_1} \bmod n, C_k := h^{r_2} \bmod n, d = h^{r_3} \bmod n, f := {C_k}^{E_k}/h^{v_k} \bmod n$ とする．

**4.** 全ての $1 \leq j \leq K$ かつ $j \neq k$ に対し，$C_j := z^{(v_i + r_2 + E_k r_1 - v_k)\prod_{\ell \neq k} E_\ell} \bmod n$ を計算する．

**5.** $e \in_R \mathbb{Z}_n^{*2}$ を選択し，$(n, d, e, f, g, h)$ をグループの公開鍵とする．

**6.** 正直なユーザが $M_G$ に証明書の発行を求めてきた場合は，$(P_j = {C_j}^{E_j}/f, E_j, C_j)$ を選ぶ．

**7.** $\mathcal{A}$ と以下のように対話をおこなう:

仮名生成プロトコル $PG$: *Step 1* では $\mathcal{A}$ からコミットメント $c_1, c_2$ を受け取り，$PK^2$ から ${c_1}^2 = (d^2)^{\tilde{r}_j}(e^2)^{\tilde{r}'_j}$ かつ ${c_2}^2 = (d^2)^{\tilde{x}_j}(e^2)^{\tilde{r}''_j}$ をみたす $\tilde{x}_j, \tilde{r}_j, \tilde{r}'_j, \tilde{r}''_j$ を抽出する．*Step 2* において，$s_j = v_j - \tilde{x}_j r_1 \in \mathbb{Z}$ を計算する．このとき $s_j \in \Delta$ である．次に，$r = s_j + (2^{\ell_\Delta} - 1) - \tilde{r}_j \bmod (2^{\ell_\Delta + 1} - 1)$ を計算し，$r$ を $\mathcal{A}$ に送る．残りはプロトコルに沿っておこなう．

証明書発行プロトコル $CI_O$: $\mathcal{A}$ から仮名 $P$ を受け取ったあと，$P = h^{v_j}$ をみたす $j$ を求める．もしこれをみたす $j$ が存在しない場合は，$CI_O$ の実行を中止する．そうでない場合は $(C_j, E_j)$ を $\mathcal{A}$ に送る．

登録証明プロトコル $CS^+$ と $CT^+$: 検証者 $V$ または検証機関 $O_j$ として実行する．それぞれの *Step 2* における $PK^2$ において，

$$(\tilde{c_1}^2/(e^2)^{\tilde{y}})^{\tilde{E}} \equiv (g^2)^{\tilde{x}}(h^2)^{\tilde{s}}(d^2)^{id}(f^2) \equiv z^{2(\tilde{x}r_1 + \tilde{s} + r_3 id + E_k r_2 - v_k)\prod_{\ell \neq k} E_\ell}$$

をみたす $\tilde{x} \in \Gamma, \tilde{s} \in \Delta, \tilde{E} \in \Lambda, \tilde{y}$ と $\tilde{c_1}$ の抽出をおこなう．

ここで，$\gcd(\tilde{E}, E_k) \neq E_k$ ならば $V$ または $O_\ell$ としてプロトコルの実行を続ける．そうでない場合は，ある $t \geq 1$ を用いて $\tilde{E} = tE_k$ とあらわすことができ，$C := (\tilde{c_1}/e^{\tilde{y}})^t/C_k \bmod n, \hat{E} := 2(\tilde{x}r_1 + \tilde{s} + r_3 id - v_k) \prod_{\ell \neq k} E_\ell$ とおくことで $(C^2)^{E_k} \equiv h^{2(\tilde{x}r_1 + \tilde{s} + r_3 id - v_k)} \equiv z^{\hat{E}} \pmod{n}$ が得られる．いま，$\tilde{E} = tE_k$ かつ $E_k\tilde{E} \in \Lambda$ より $1 \leq t \leq 2^{\ell_\Sigma}$ であり，$0 < 2|\tilde{x}r_1 + \tilde{s} + r_3 id - v_k| < 2(2^{2\ell_\Gamma} + 2^{\ell_\Gamma} + 2^{2\ell_\Delta} + 2^{\ell_\Delta}) < 2^{\ell_\Lambda}$ より $\gcd(E_k, \hat{E}) = 1$ を得る．拡張ユークリッドの互助法を用いて $\alpha E_k + \beta\tilde{E} = 1$ をみたす $\alpha, \beta \in \mathbb{Z}$ を求め，$u := z^\alpha C^\beta \bmod n, E := E_k$ とすれば，これらは $u^E \equiv z \pmod{n}$ をみたす．この $(u, E)$ は強 RSA 問題の解であり，これらを出力して停止する．

**8.** もし $\mathcal{A}$ が停止したならば，Null を返し停止する．

これまでの議論から，以下の補題，及び定理を得ることができる．

**補題 5** 実際のプロトコルにおいて攻撃者の得る情報はシミュレーションで得る情報と統計的識別不可能である．

**定理 6** 強 RSA 問題仮定と Diffie-Hellman 決定問題仮定，素因数分解問題仮定のもと，提案する匿名証明書方式は安全である．

### 5.3.5 不正なユーザの登録削除

匿名証明書方式では，ユーザの匿名性が守られているため，機関との取引において不正を行ったユーザの匿名性をも保証することになる．まず，このようなユーザの削除 (追跡) を行うための機関 $RM$ を用意し，これを不正管理者とよぶことにする．以下に不正なユーザの削除を可能にするためのプロトコルを与える．但し，ここでは基本的プロトコルに追加される部分のみを述べることにする．

#### 鍵生成プロトコル

登録管理者 $RM$ は，素数位数 $q > 2^{\ell_\Gamma}$ をもつ群 $G = \langle g_R \rangle = \langle h_R \rangle$ を選択する．さらに秘密鍵 $x_1, x_2, x_3, x_4, x_5$ を選択し，$(y_1, y_2, y_3) := ({g_R}^{x_1}{h_R}^{x_2}, {g_R}^{x_3}{h_R}^{x_4}, {g_R}^{x_5})$

をそれに対応する公開鍵とし，ハッシュ関数 $\mathcal{H} : \{0,1\}^* \to \{0,1\}^q$ とともに公開する．

また各機関管理者 $M_G$ は，生成元 $v_G \in \mathbb{Z}_{{n_G}^2}$ を公開鍵に加える．

### 仮名生成プロトコルの変更

ユーザの機関 $O \in \boldsymbol{G}$ における仮名 $P_{(U,O)}$ を $U$ の新たな秘密情報 $x_{(U,O)} \in \Gamma$ を用いた $P_{(U,O)} = {g_G}^{x_U} {h_G}^{s_{(U,O)}} {v_G}^{x_{(U,O)}}$ に変更し，それに対応する登録証明書は ${C_{(U,O)}}^{E_{(U,O)}} \equiv P_{(U,O)} f_G \pmod{n_G}$ をみたす $(E_{(U,O)}, C_{(U,O)})$ の組で与えられる．それに伴い，機関 $O$ との仮名生成プロトコル $PG$ を以下のように変更する．

*Step 1.* $U$ は $r_1 \in_R \Delta, r_2, r_3 \in_R \{0,1\}^{2\ell_n}$ に対し，$c_1 := {d_G}^{r_1} {e_G}^{r_2}, c_2 := {d_G}^{x_U} {e_G}^{r_3}$ を計算し，それらを $Y_{(U,O)} = {g_R}^{x_{(U,O)}}$ とともに $O$ に送る．

$$PK^2\{(\alpha,\beta,\gamma,\delta) \ : \ c_1 = {d_G}^{\alpha} {e_G}^{\beta} \wedge c_2 = {d_G}^{\gamma} {e_G}^{\delta}\}$$

によって，$c_1, c_2$ が正しく生成されていることを証明する．

*Step 2.* $O$ は $r \in_R \Delta$ を選択し，$U$ に送信する．

*Step 3.* $U$ は，$s_{(U,O)} := (r_1 + r \pmod{2^{\ell_\Delta+1}+1}) - 2^{\ell_\Delta} + 1$ と $\tilde{s} = \left\lfloor \frac{r_1+r}{2^{\ell_\Delta+1}-1} \right\rfloor$ を計算し，$P_{(U,O)} := {g_G}^{x_U} {h_G}^{s_{(U,O)}} {v_G}^{x_{(U,O)}}$ を仮名とする．また $r_4 \in_R \{0,1\}^{\ell_n}$ に対し $c_3 := {d_G}^{\tilde{s}} {e_G}^{r_4}$ と $P_{(U,O)}$ を $X$ に送信し，それが正しく生成されたことを以下の $PK^2$ によって証明する:

$$\begin{aligned} PK^2\{(\alpha,\beta,\gamma,\delta,\varepsilon,\zeta,\vartheta,\xi) \quad : \quad & c_1 = {d_G}^{\alpha} {e_G}^{\beta} \\ \wedge \quad & c_2 = {d_G}^{\gamma} {e_G}^{\delta} \\ \wedge \quad & c_3 = {d_G}^{\varepsilon} {e_G}^{\zeta} \\ \wedge \quad & P_{(U,O)} = {g_G}^{\gamma} {h_G}^{\vartheta} {v_G}^{\phi} \\ \wedge \quad & ({c_1} {d_G}^{r-2^{\ell_\Delta+1}}) / ({c_3}^{2^{\ell_\Delta+1}+1}) = {d}^{\vartheta} {e}^{\xi} \\ \wedge \quad & Y_{(U,O)} = {g_R}^{\phi} \\ \wedge \quad & \gamma \in \Gamma \wedge \vartheta \in \Delta, \phi \in \Gamma\}. \end{aligned}$$

*Step 4.* $O$ は仮名リストに $P_{(U,O)}$ を保管する．

*Step 5.* $U$ は $P_{(U,O)}$ の生成に用いた秘密情報 $s_{(U,O)}$ と $P_{(U,O)}$ を $O$ の登録情報として秘密に保管する．

### 登録証明プロトコルの変更

登録証明プロトコル $CS^+$ に以下のステップを加える．

*Step 3.* ユーザ $U$ は，$r_3 \in \mathbb{Z}_n$ を選択し，$w_1 := {g_R}^{r_3}, w_2 := {h_R}^{r_3}, w_3 := {Y_{(U,O)}y_3}^{r_3}$ を計算する．また，$e := \mathcal{H}(w_1, w_2, w_3)$ に対して $w_4 := (y_1{y_2}^e)^{r_3}$ とし，$(w_1, w_2, w_3, w_4)$ を検証者 $V$ に送る．

*Step 4.* $U$ は以下の $PK^2$ によって，$V$ に登録削除に必要な情報が正しく生成されていることを証明する．

$$
\begin{aligned}
PK^2\{(\alpha,\beta,\gamma,\delta,\varepsilon,\xi) \ : \quad & f_G{d_G}^{id_{(O_i,G)}} = {c_1}^{\alpha}/g^{\beta}h^{\gamma}v^{\delta}e^{\varepsilon} \\
\wedge \quad & w_1 = {g_R}^{\xi} \\
\wedge \quad & w_2 = {h_R}^{\xi} \\
\wedge \quad & w_3 = {g_R}^{\delta}{y_3}^{\xi} \\
\wedge \quad & w_4 = \left(y_1{y_2}^{\mathcal{H}(w_1,w_2,w_3)}\right)^{\varepsilon}\}.
\end{aligned}
$$

また，登録証明をおこなう他のプロトコル $CS^-, CT^+, CT^-$ においても同様にステップの追加をおこなう．

### ユーザの削除

ユーザの不正が発覚した場合，検証者 $V$ は $(w_1, w_2, w_3, w_4)$ を不正管理者 $RM$ に提出する．$RM$ は，$\tilde{e} := \mathcal{H}(w_1, w_2, w_3)$ に対して $w_4 = {w_1}^{x_1+x_3\tilde{e}}{w_2}^{x_2+x_4\tilde{e}}$ がなりたつかどうかを検証する．これらが正しければ，$\tilde{Y} := w_3/{w_1}^{x_5}$ を計算し，対応する機関に $PK\{(\alpha) : w_3/\tilde{Y} = {w_1}^{\alpha}\}$ の証明とともに $\tilde{Y}$ を送る．機関 $O$ は $\tilde{Y}$ に対応するユーザを削除し公開する．

## 5.4 複数匿名証明書方式

複数機関への登録を効率よく示すことのできる複数匿名証明書方式は以下のエンティティによって構成される:

証明書発行機関 $IO$: ユーザに証明書を発行するエンティティ.

機関 $O_i$: グループに属するエンティティ.

ユーザ $U$: グループに登録し，そのグループ内の機関と取引をおこなうエンティティ.

検証者 $V$: ユーザの機関への登録を検証するエンティティ.

### 5.4.1 提案方式の概要

複数匿名証明書方式の流れについて述べる.

1. 初期設定:

   登録証明書発行機関 $IO$，登録認証機関 $O_i$，ユーザ $U$ は鍵生成プロトコル $KG$ を用いて，それぞれ秘密鍵と公開鍵を生成する．$IO$ は RSA の法 $n$ と $d, e, f, g, h, v \in \mathbb{Z}_n^{*2}$ を公開鍵とし，$O_i$ は秘密鍵 $x_{O_i}$ に対する $y_{O_i} := v^{x_{O_i}} \bmod n$ を公開鍵とする．また，ユーザ $U$ は，秘密鍵 $x_U$ と証明書発行の際に用いる情報 $E_U$ を生成する.

2. システムへの登録:

   ユーザ $U$ と証明書発行機関 $IO$ は，仮名生成プロトコル $PG$ によって $P_{(U,O_i)}$ を生成し，$IO$ での仮名として $P_{(U,IO)}$ を登録する．これは $U$ の秘密鍵 $x_U$ を用いて $P_{(U,IO)} := g^{x_U} h^{s_{(U,IO)}} \bmod n$ で生成される．但し，$s_{(U,IO)}$ はプロトコル $PG$ によって生成される $U$ の秘密情報である.

3. 登録証明書の発行:

   3-1. ユーザ $U$ は機関 $O_i$ に，仮名生成プロトコル $PG$ を用いて生成した $P_{(U,O_i)} := g^{x_U} h^{s_{(U,O_i)}} \bmod n$ を登録する.

3-2. ユーザ $U$ と機関 $O_i$ は，仮名の保証書生成プロトコル $GG$ によって，$U$ の $O_i$ への登録を証明書発行機関 $IO$ に保証するための情報である $\sigma_{(U,O_i)}$ を発行する．$\sigma_{(U,O_i)}$ は $P_{(U,O_i)}$ に対して生成され，$U$ へのプライベート出力である．

3-3. 登録証明書発行プロトコル $CI$ によって，証明書発行機関 $IO$ は $\sigma_{(U,O_i)}$ を用いて登録証明書 $C_{(U,O_i)}$ を発行する．また，$IO$ に対するユーザの匿名性を強化するため，ユーザ $U$ は登録先の機関の情報を $IO$ に与えない．プロトコル $IO$ において，$IO$ は $\sigma_{(U,O_i)}$ の生成機関がどれであるのかわからないが，確かに $\sigma_{(U,O_i)}$ は $P_{(U,IO)}$ と同じ秘密鍵を用いて生成された仮名に対して発行されていることを確認し，その機関の公開鍵を用いて正しい登録証明書を発行することができる．この証明書は $O_i$ の公開鍵 $y_{O_i}$ に対して $C_{(U,O_i)} \equiv ({y_{O_i}}^{x_U} f)^{1/E_U} \pmod{n}$ をみたす $C_{(U,O_i)}$ で与えられる．但し，$E_U$ は，$U$ の全ての登録証明書の生成に用いられる共通の値である．

4. 登録証明: ユーザは検証者 $V$ もしくは機関 $O_j$ に登録証明をおこなう．ユーザ $U$ が登録証明書の発行を受けた機関の集合を $\mathcal{O}_U$ とし，検証者 $V$ もしくは $O_j$ に証明したい登録機関の集合を $\mathcal{O}_S \subset \mathcal{O}_U$ とする．

– 登録証明プロトコル $CS$ では，$C_U := \prod_{\mathcal{O}_S} C_{(U,O_i)}$ に対して，

$$ {C_U}^{E_U} \equiv (\prod_{\mathcal{O}_S} y_{O_i})^{x_U} f^{|\mathcal{O}_S|} \pmod{n} $$

をみたす，$C_U, E_U, x_U$ の知識証明をおこなうことで，検証者 $V$ に $O_i \in \mathcal{O}_S$ への登録証明をおこなうことができる．

– 検証機関に対する属性証明プロトコル $CT$ によって，機関 $O_j$ に対し，$C_U := \prod_{\mathcal{O}_S} C_{(U,O_i)}$ を用いて，

$$ {C_U}^{E_U} \equiv (\prod_{\mathcal{O}_S} y_{O_i})^{x_U} f^{|\mathcal{O}_S|} \pmod{n} $$
$$ P_{(U,O_j)} = g^{x_U} h^{s_{(U,O_j)}} \pmod{n} $$

を同時にみたす，$C_U, E_U, x_U, s_{(U,O_j)}$ の知識証明をおこなうことで，$P_{(U,O_j)}$ に対応するユーザの $O_i \in \mathcal{O}_S$ への登録を証明することができる．

### 5.4.2 プロトコル

本章では，効率よく複数機関への登録を証明するためのプロトコルを提案する．

#### 初期設定

システムパラメータを以下のように設定する．RSA の法のサイズを $\ell_n$ ビット，セキュリティパラメータを $\epsilon > 1$ する．また，$\ell_\Gamma = 2\ell_n$, $\ell_\Delta = \epsilon\ell_\Gamma$, $2^{\ell_\Lambda} > 2(2^{\ell_\Gamma+2})$ とし，各区間を $\Gamma = ]-2^{\ell_\Gamma}, 2^{\ell_\Gamma}[$, $\Delta = ]-2^{\ell_\Delta}, 2^{\ell_\Delta}[$, $\Lambda = ]2^{\ell_\Lambda}, 2^{\ell_\Lambda+\ell_\Sigma}[$ と設定する．

#### 鍵生成プロトコル $KG$

鍵生成プロトコル $KG_{IO}, KG_{O_i}$ と $KG_U$ は

$$KG_{IO}(1^k) \ni (\mathcal{X}_{IO}, \mathcal{Y}_{IO}), KG_{O_i}(1^k, \mathcal{Y}_{IO}) \ni (x_O, y_O), KG_U \ni x_U$$

であらわされ，$KG_{IO}$ と $KG_{O_i}$ は，それぞれ証明書発行機関 $IO$ と機関 $O_i$ の秘密鍵, 公開鍵の組を，$KG_U$ はユーザ $U$ の秘密鍵を生成する．$(\mathcal{X}_{IO}, \mathcal{Y}_{IO})$ を $IO$ の鍵，$(x_{O_i}, y_{O_i})$ を機関 $O_i$ の鍵，そして $x_U$ をユーザ $U$ の秘密鍵とする．上記は，$KG_{IO}$ と $KG_U$ は $1^k$ の入力に対し，$(\mathcal{X}_{IO}, \mathcal{Y}_{IO})$ または $x_U$ を，$KG_{O_i}$ は $1^k$ と $IO$ の公開鍵 $\mathcal{Y}_{IO}$ に対し，$(x_{O_i}, y_{O_i})$ を出力することを意味する．

*Step 1.* 証明書発行機関 $IO$ は，$p := 2p' + 1, q = 2q' + 1$ を素数とする $\ell_n/2$ ビットの素数 $p', q'$ をランダムに選び，$n := pq$ とする．また，$d, e, f, g, h, v \in_R \mathbb{Z}_n^{*2}$ を選び，$(p, q)$ を秘密鍵，$(n, d, e, f, g, h, v)$ を公開鍵とする．

*Step 2.* 属性認証機関 $O_i$ は，秘密鍵 $x_{O_i} \in_R \Gamma$ を選択し，それに対応する公開鍵を $y_{O_i} := v^{x_{O_i}} \bmod n$ とする．

*Step 3.* ユーザ $U$ は，秘密鍵として $x_U \in \Gamma$ と，証明書生成に用いる情報として素数 $E_U \in \Lambda$ を選択し，保管しておく．

**仮名生成プロトコル** $PG$

仮名生成プロトコル $PG$ は,

$$PG\langle U(x_U), X\rangle(\mathcal{Y}_{IO}) \ni [U(s_{(U,X)})](P_{(U,X)})$$

であらわされる．上記は，$PG$ は $U$ と $X \in \{O, IO\}$ の対話によっておこなわれ，これは，$U$ による入力 $x_U$ と共通入力 $\mathcal{Y}_{IO}$ から，$U$ へ $s_{(U,X)}$ を $U$ へプライベート出力し，$U$ の仮名 $P_{(U,X)}$ を共通出力とすることを意味する．また，$PG$ は出力 $P_{(U,X)}$ が正しい型であること，即ち $x_U \in \Gamma, s_{(U,X)} \in \Delta$ を用いて $P_{(U,X)} = g^{x_U} h^{s_{(U,X)}}$ にあらわされることを保証する．

プロトコル $PG$ は以下のように構成される:

*Step 1.* $U$ は $r_1 \in_R \Delta, r_2, r_3 \in_R \{0,1\}^{2\ell_n}$ を選び，$c_1 := d^{r_1} e^{r_2}, c_2 := d^{x_U} e^{r_3}$ を計算し，$X$ に送る．また,

$$PK^2\{(\alpha, \beta, \gamma, \delta) \ : \ c_1 = d^{\alpha} e^{\beta} \wedge c_2 = d^{\gamma} e^{\delta}\}$$

によって，$c_1, c_2$ が正しく生成されていることを証明する．

*Step 2.* $X$ は $r \in_R \Delta$ を選択し，$U$ に送信する．

*Step 3.* $U$ は，$s_{(U,X)} := (r_1 + r \pmod{2^{\ell_\Delta+1} + 1}) - 2^{\ell_\Delta} + 1$ と $\tilde{s} = \lfloor r_1 + r/(2^{\ell_\Delta+1} - 1) \rfloor$ を計算し，$P_{(U,X)} := g^{x_U} h^{s_{(U,X)}}$ を仮名とする．また $r_4 \in_R \{0,1\}^{\ell_n}$ に対し $c_3 := d^{\tilde{s}} e^{r_4}$ と $P_{(U,X)}$ を $X$ に送信し，それが正しく生成されたことを以下のゼロ知識証明によって証明する．

$$\begin{aligned} PK^2\{(\alpha, \beta, \gamma, \delta, \varepsilon, \zeta, \vartheta, \xi) \ : \ & c_1 = d^{\alpha} e^{\beta} \\ \wedge \ & c_2 = d^{\gamma} e^{\delta} \\ \wedge \ & c_3 = d^{\varepsilon} e^{\zeta} \\ \wedge \ & P_{(U,X)} = g^{\gamma} h^{\vartheta} \\ \wedge \ & (c_1 d^{r - 2^{\ell_\Delta+1}}) / ({c_3}^{2^{\ell_\Delta+1}+1}) = d^{\vartheta} e^{\xi} \\ \wedge \ & \gamma \in \Gamma \wedge \vartheta \in \Delta\}. \end{aligned}$$

*Step 4.* $X$ は仮名リストに $P_{(U,X)}$ を保管する．

*Step 5.* $U$ は $P_{(U,X)}$ の生成に用いた秘密情報 $s_{(U,X)}$ と $P_{(U,X)}$ を $X$ の登録情報として秘密に保管する．

**仮名の保証書生成プロトコル** $GG$

仮名生成プロトコル $GG$ によってユーザは仮名を登録した機関 $O_i$ から登録済みであることを保証する情報を得ることができる．このプロトコルは，

$$GG\langle U(x_U, s_{(U,O)}), O_i(x_{O_i})\rangle(\mathcal{Y}_{IO}, y_{O_i}, P_{(U,O_i)}) \ni [U(\sigma_{(U,O_i)}, r_{(U,O_i)})]$$

であらわされ，$U$ のプライベート入力 $x_U$ に対し，$P_{(U,O_i)} \in PG\langle U(x_U), O_i\rangle$ のときのみ $U$ の仮名 $P_{(U,O_i)}$ に対する仮名証明書 $\sigma_{(U,O_i)}$ をプライベート出力として $U$ に与える．

プロトコル $GG$ は以下のように構成される:

*Step 1.* $U$ は $O_i$ に登録済みの仮名 $P_{(U,O_i)}$ に対応するユーザであることを

$$PK^2\{(\alpha, \beta) : P_{(U,O_i)} = g^\alpha h^\beta\}$$

によって証明する．

*Step 2.* $O_i$ は $r \in_R \{0,1\}^{2\ell_n}$ に対して，$t_1 := v^r, t_2 = {P_{(U,O_i)}}^r$, $Q_{(U,O_i)} := {P_{(U,O_i)}}^{x_{O_i}}$ を計算し，それらを $U$ に送る．

*Step 3.* $U$ は $r_1, r_2, r_{(U,O_i)} \in \{0,1\}^{2\ell_n}$ を選択し，$\tilde{t_1} := t_1 v^{r_1} {y_{O_i}}^{r_2}$, $\tilde{t_2} := (t_2 {P_{(U,O_i)}}^{r_1} {Q_{(U,O_i)}}^{r_2})^{r_{(U,O_i)}}, \tilde{P} := {P_{(U,O_i)}}^{r_{(U,O_i)}}, \tilde{Q} := {Q_{(U,O_i)}}^{r_{(U,O_i)} x_U}$ を計算する．

次に，$\tilde{e} := \mathcal{H}(\tilde{t_1}, \tilde{t_2})$, $e := \tilde{e} x_U - r_2$ を計算し，$e$ を $O_i$ に送る．

*Step 4.* $O_i$ は秘密鍵 $x_{O_i}$ を用いて，$s := r - e x_{O_i}$ を生成し，$U$ に送る．

*Step 5.* $U$ は $t_1 = v^s {y_{O_i}}^e$, $t_2 = {P_{(U,O_i)}}^s {Q_{(U,O_i)}}^e$ が成り立つかどうかを検証し，それが正しければ，$\tilde{s} := s + r_1$ とし，$P_{(U,O_i)}$ に対する登録証明書が発行されるまで，$\sigma_{(U,O_i)} := (\tilde{s}, \tilde{t_1}, \tilde{t_2}, \tilde{P}, \tilde{Q})$ と $r_{(U,O_i)}$ を保管する．

**登録証明書発行プロトコル** $CI$

$CI$ は，証明書発行機関 $IO$ が登録証明書を発行するのに用いられ，

$$CI\langle U(x_U, s_{(U,IO)}, s_{(U,O_i)}, P_{(U,O_i)}, r_{(U,O_i)}, y_{O_i}), IO(\mathcal{X}_{IO})\rangle(\mathcal{Y}_{IO}, P_{(U,IO)}, \sigma_{(U,O_i)}) \ni \mathcal{C}_{(U,IO)}$$

であらわされる．$CI$ は $U$ のプライベート入力 $x_U, P_{(U,O_i)}, y_{O_i}$ に対し，$\sigma_{(U,O_i)} \in GG\langle U(x_U), O_i\rangle(P_{(U,O_i)})$ かつ $P_{(U,IO)} \in PG\langle U(x_U), IO\rangle$ であるとき，$C_{(U,O_i)} \equiv ({y_{O_i}}^{x_U} f)^{1/E_U} \pmod n$ をみたす登録証明書 $C_{(U,O_i)}$ を出力する．$CI$ は $\log_v \tilde{y} = \log_{\tilde{P}} \tilde{Q} = x_U$ をみたす $\tilde{y}$ に対して発行されるため，$\tilde{y}^{1/x_U}$ を公開鍵としてもつ機関への登録証明書となる．

また，$CI$ において $U$ は $\sigma_{(U,O_i)}$ に対応するユーザであること，即ち $\tilde{P}$ に対応する仮名が $P_{(U,IO)}$ と同じ秘密鍵を用いて生成されていることのみを証明し，$\sigma_{(U,O_i)}$ の作成機関に関する情報は何ら与えない．

$U$ と $IO$ は以下のように構成される:

*Step 1.* $U$ は，$\sigma_{(U,O_i)}$ を $IO$ に送信し，$\sigma_{(U,O_i)}$ に対応するユーザであることを証明するため，以下の対話証明をおこなう:

$$\begin{aligned} PK^2\{(\alpha,\beta,\gamma,\delta,\varepsilon,\zeta) \quad : \quad & P_{(U,IO)} = g^\alpha h^\beta \\ \wedge \quad & \tilde{P} = g^\gamma h^\delta \\ \wedge \quad & 1 = {P_{(U,IO)}}^\varepsilon / (g^\gamma h^\zeta) \\ \wedge \quad & \alpha \in \Gamma, \beta \in \Delta\}. \end{aligned}$$

*Step 2.* $IO$ は $\tilde{e} = \mathcal{H}(\tilde{t}_1, \tilde{t}_2)$ に対して，$\tilde{t}_2 \equiv \tilde{P}^{\tilde{s}} \tilde{Q}^{\tilde{e}}$ が成り立つかどうかを検証し，正しければ $\tilde{y} := (\tilde{t}_1 / v^{\tilde{s}})^{1/\tilde{e}}$ に対し，$C_{(U,O_i)} := (\tilde{y} f)^{1/E_U} \bmod n$ とし $U$ に送る．

*Step 3.* $U$ は，${C_{(U,O_i)}}^{E_U} \equiv {y_{O_i}}^{x_U} f \pmod n$ が成り立つかどうか検証する．もし正しければ，$C_{(U,O_i)}$ を $O_i$ の登録証明書として保管する．

### 登録証明プロトコル $CS$

$CS$ は，ユーザ $U$ が検証者 $V$ に任意の複数機関への登録を証明するのに用いられる．ここで，$U$ が既に証明書の発行を受けている機関の集合を $\mathcal{O}_U$，$V$ に証明する機関の集合を $\mathcal{O}_S \subset \mathcal{O}_U$ とすると，$CS$ は，

$$CS\langle U(x_U, \{\mathcal{C}_{(U,G)}\}_{\mathcal{O}_S}), V\rangle(\mathcal{Y}_{IO}, \{y_O\}_{\mathcal{O}_S}) \ni \{0,1\}$$

であらわされ，$\{C_{(U,O)} \in CI\langle U(x_U, y_O), IO\rangle\}_{\mathcal{O}_S}$ の入力に対して圧倒的確率で 1 を，そうでないときは 0 を出力する．

プロトコル $CS$ は以下のように構成される:

*Step 1.* $U$ は $C_U := \prod_{\mathcal{O}_s} C_{(U,O_i)}$ を計算し，$r_1, r_2 \in_R \{0,1\}^{2\ell_n}$ に対し，$c_1 := C_U e^{r_1}$，$c_2 := e^{r_1} d^{r_2}$ を生成し，検証者 $V$ に送信する．

*Step 2.* $U$ と $V$ は以下の $PK$ を実行する:

$$\begin{aligned} PK^2\{(\alpha,\beta,\gamma,\delta,\varepsilon,\zeta) \quad : \quad & f^{|\mathcal{O}_S|} = {c_1}^{\alpha}/((\textstyle\prod_{\mathcal{O}_S} y_{O_i})^{\beta} e^{\gamma}) \\ \wedge \quad & c_2 = e^{\delta} d^{\varepsilon} \\ \wedge \quad & 1 = {c_2}^{\alpha}/e^{\gamma} d^{\zeta} \\ \wedge \quad & \alpha \in \Lambda \wedge \beta \in \Gamma\}. \end{aligned}$$

**検証機関に対する登録証明プロトコル** $CT$

$CT$ は，ユーザ $U$ が検証機関 $O_j$ に $U$ の全ての属性 $\mathcal{O}_S$ に対する $\mathcal{O}_S \subset \mathcal{O}_U$ への登録を示すとともに，$O_j$ における $P_{(U,O_j)}$ に対応するユーザであることを証明するのに用いられる．

$$CT\langle U(x_U, \{\mathcal{C}_{(U,G_I)}\}_{\mathcal{O}_S}), O_j\rangle(\mathcal{Y}_{IO}, \{y_O\}_{\mathcal{O}_S} P_{(U,O_j)}) \ni \{0,1\}$$

であらわされ，証明者 $U$ が $\{\mathcal{C}_{(U,O)} \in CI\langle U, IO\rangle(P_{(U,O)}), P_{(U,O)} \in PG\langle U(x_U), O\rangle\}_{\mathcal{O}_S}$ かつ $P_{(U,O_j)} \in PG\langle U(x_U), O_j\rangle$ をみたす $\{C_{(U,O)}\}_{\mathcal{O}_S}$ を入力したとき，圧倒的確率で 1 を，そうでないときは 0 を出力する．

プロトコル $TC$ は以下のように構成される:

*Step 1.* $U$ は $C_U := \prod_{\mathcal{O}_s} C_{(U,O_i)}$ を計算し，$r_1, r_2 \in_R \{0,1\}^{2\ell_n}$ に対し，$c_1 := C_U e^{r_1}$，$c_2 := e^{r_1} d^{r_2}$ を生成し，機関 $O_j$ に送信する．

*Step 2.* $U$ と $O_j$ は，$O_j$ に登録されている仮名 $P_{(U,O_j)}$ に対して以下の $PK$ を実行する:

$$\begin{aligned} PK^2\{(\alpha,\beta_i,\gamma,\delta,\varepsilon,\zeta,\xi,\eta,\varphi) \quad : \quad & f^{|\mathcal{O}_S|} = {c_1}^{\alpha}/((\textstyle\prod_{\mathcal{O}_S} y_{O_i})^{\beta} e^{\gamma}) \\ \wedge \quad & c_2 = e^{\delta} d^{\varepsilon} \\ \wedge \quad & 1 = {c_2}^{\alpha}/e^{\gamma} d^{\zeta} \\ \wedge \quad & P_{(U,O_j)} = g^{\beta} h^{\xi} \\ \wedge \quad & \alpha \in \Lambda \wedge \beta \in \Gamma \wedge \xi \in \Delta\}. \end{aligned}$$

### 5.4.3 安全性の考察

本章では，提案する匿名証明書方式の安全性について考察する．強 RSA 問題仮定，Diffie-Hellman 決定問題仮定の下，各プロトコルにおいて以下の補題を与えることができる．

**補題 6** *CI* における

$$\begin{aligned} PK^2\{(\alpha,\beta,\gamma,\delta,\varepsilon,\zeta) \quad : \quad & P_{(U,IO)} = g^{\alpha}h^{\beta} \\ \wedge \quad & \tilde{P} = g^{\gamma}h^{\delta} \\ \wedge \quad & 1 = {P_{(U,IO)}}^{\varepsilon}/(g^{\gamma}h^{\zeta}) \\ \wedge \quad & \alpha \in \Gamma, \beta \in \Delta\} \end{aligned}$$

は，$P_{(U,IO)}$ と $\tilde{P}$ が同じ鍵を用いて生成されていることのゼロ知識証明である．

*Proof.* $PK^2$ プロトコルの性質より，述語部分をみたす $\alpha,\beta,\gamma,\delta,\varepsilon,\zeta$ を抽出することができる．また，$P_{(U,IO)} = g^{\alpha}h^{\beta}, \tilde{P} = g^{\gamma}h^{\delta} \wedge 1 = {P_{(U,IO)}}^{\varepsilon}/(g^{\gamma}h^{\zeta})$ より，$\alpha\varepsilon \equiv \gamma \pmod{ord(g)}$ を得る．よって，$\tilde{P} = g^{\gamma}h^{\delta} = (g^{\alpha}h^{\delta/\varepsilon})^{\varepsilon}$ より，$\tilde{P}$ は $P_{(U,IO)}$ と同じ鍵を用いて生成されていることがいえる．

**補題 7** *CS* における

$$\begin{aligned} PK^2\{(\alpha,\beta,\gamma,\delta,\varepsilon,\zeta) \quad : \quad & f^{|\mathcal{O}_S|} = {c_1}^{\alpha}/((\textstyle\prod_{\mathcal{O}_S} y_{O_i})^{\beta}e^{\gamma}) \\ \wedge \quad & c_2 = e^{\delta}d^{\varepsilon} \\ \wedge \quad & 1 = {c_2}^{\alpha}/e^{\gamma}d^{\zeta} \\ \wedge \quad & \alpha \in \Lambda \wedge \beta \in \Gamma\} \end{aligned}$$

は，$IO$ によって発行された $\mathcal{O}_S \ni O_i$ の正しい属性証明書の統計的ゼロ知識証明である．

*Proof.* $PK^2$ プロトコルの性質より，述語部分をみたす $\alpha,\beta,\gamma,\delta,\varepsilon,\zeta$ を抽出することができる．また，$c_2 = e^{\delta}d^{\varepsilon}, 1 = {c_2}^{\alpha}/e^{\gamma}d^{\zeta}$ より，$\alpha\delta \equiv \gamma \pmod{ord(e)}$ を得る．よって，$f^{|\mathcal{O}_S|} = {c_1}^{\alpha}/((\prod_{\mathcal{O}_S} y_{O_i})^{\beta}e^{\gamma})$ より，${c_1}^{\alpha}/e^{\gamma} = (c_1/e^{\delta})^{\alpha} = (\prod_{\mathcal{O}_S} y_{O_i})^{\beta}f^{|\mathcal{O}_S|}$．このことから，証明者は $IO$ によって発行された正しい証明書を保持していることがいえる．

**補題 8** *CT* における

$$
\begin{aligned}
PK^2\{(\alpha,\beta_i,\gamma,\delta,\varepsilon,\zeta,\xi,\eta,\varphi) \quad : \quad & f^{|\mathcal{O}_S|} = {c_1}^{\alpha}/((\textstyle\prod_{\mathcal{O}_S} y_{O_i})^{\beta} e^{\gamma}) \\
\wedge \quad & c_2 = e^{\delta} d^{\varepsilon} \\
\wedge \quad & 1 = {c_2}^{\alpha}/e^{\gamma} d^{\zeta} \\
\wedge \quad & P_{(U,O_j)} = g^{\beta} h^{\xi} \\
\wedge \quad & \alpha \in \Lambda \wedge \beta \in \Gamma \wedge \xi \in \Delta\}
\end{aligned}
$$

は，$P_{(U,O_j)}$ に対応するユーザによる $IO$ によって発行された $\mathcal{O}_S \ni O_i$ の正しい登録証明書の統計的ゼロ知識証明である，

*Proof.* $PK^2$ プロトコルの性質より，述語部分をみたす $\alpha,\beta,\gamma,\delta,\varepsilon,\zeta$ を抽出することができる．また，$c_2 = e^{\delta} d^{\varepsilon}, 1 = {c_2}^{\alpha}/e^{\gamma} d^{\zeta}$ より，$\alpha\delta \equiv \gamma \pmod{ord(e)}$ を得ることができ，$f^{|\mathcal{O}_S|} = {c_1}^{\alpha}/((\prod_{\mathcal{O}_S} y_{O_i})^{\beta} e^{\gamma})$ より，${c_1}^{\alpha}/e^{\gamma} = (c_1/e^{\delta})^{\alpha} = (\prod_{\mathcal{O}_S} y_{O_i})^{\beta} f^{|\mathcal{O}_S|}$，かつ $P_{(U,O_j)} = g^{\beta} h^{\zeta}$ より，証明者は $IO$ によって発行された正しい証明書を保持しており，かつ $P_{(U,O_j)}$ は同じ鍵を用いて生成されているといえる．

### 5.4.4 シミュレータの構築

提案する匿名証明書方式の安全性を示すためにシミュレータを構築し，それが安全な属性認証システムの定義を満たすことを示す必要がある．ここで，攻撃者 $\mathcal{A}$ がコントロールするエンティティは一つのエンティティに包摂されるものとし，$\mathcal{A}$ のためのシミュレータの構築を目指す．

**初期設定**

$\mathcal{A}$ にコントロールされない証明書発行機関 $IO$ と機関 $O$ において，$\mathcal{S}$ はそれらの秘密鍵と公開鍵の組 $(\mathcal{X}_{IO}, \mathcal{Y}_{IO})$ と $(x_O, y_O)$ を用意する．さらに，$\mathcal{S}$ は $\mathcal{A}$ によってコントロールされたユーザの登録を記録する $archive_{IO}$ と $archive_O$ を作成する．$archive_{IO}$ は $IO$ に仮名を登録したユーザ，$archive_O$ は機関 $O$ の属性証明書の発行をうけたユーザのデータをそれぞれ記録するものとする．さらに，$\mathcal{A}$ によってコントロールされるユーザのリスト $list_{\mathcal{A}}$ を初期化しておく．

**証明書発行機関への仮名登録**

ユーザは証明書発行機関 $IO$ に登録する仮名を生成する:

(I) $\mathcal{A}$ によってコントロールされるユーザが正直な証明書発行機関に仮名を登録する場合，(i) $\mathcal{S}$ は，プロトコル $PG$ における知識抽出によって，ユーザの秘密鍵 $x$ と秘密情報 $s$ を得る．(i-1) その秘密鍵 $x$ に対応するユーザが $list_{\mathcal{A}}$ に存在しなければ，$\mathcal{S}$ はログインネームを $L_U$ とする新しいユーザを作成し，$T$ と対話することによって仮名 $N_{(U,G)}$ と，$L_U$ に対応する鍵 $K_U$ を手に入れる．さらに，$(x_U := x, s_{(U,G)} := s)$ とし，$\mathcal{S}$ は $\mathcal{A}$ にコントロールされたユーザのリスト $list_{\mathcal{A}}$ に $(U, L_U, x_U, K_U, N_{(U,G)}, s_{(U,G)})$ を加える．(ii-2) $x$ に対応するユーザが $list_{\mathcal{A}}$ に既に存在するならば，$\mathcal{S}$ は，$T$ と対話することによって $x$ に対応するユーザ $U$ のために仮名 $N_{(U,G)}$ を手に入れ，$U$ の情報に $N_{(U,G)}, s_{(U,G)} := s$ を加える．

(II) $T$ を通して正直なユーザが $\mathcal{A}$ によってコントロールされたグループ管理者に登録する仮名を作成する場合，$\mathcal{S}$ はプロトコル $TC$ におけるゼロ知識シミュレータを用いて，$\mathcal{A}$ の動きをシミュレートする．

**機関への仮名登録**

ユーザは機関 $O$ に登録する仮名を生成する:

このシミュレーションは上のグループへの仮名登録と同様におこなうことができる．

**仮名保証書の生成**

ユーザは仮名の保証書の生成を機関 $O$ に要求する:

(I) ユーザが $\mathcal{A}$ によってコントロールされている場合，(i) $\mathcal{S}$ はユーザの鍵 $x$ と秘密情報 $s$ を得るためにプロトコル $GV$ の *Step 1* の知識抽出をおこなう．

$(x, s)$ に対応する $N$ に対して, (i-1) $N \notin list_{\mathcal{A}}$ ならば，$\mathcal{S}$ は保証書の発行を拒否する．(i-2) $N \in list_{\mathcal{A}}$ ならば，$\mathcal{S}$ は $T$ との対話によって $\sigma$ を作成する．

(II) 機関 $O$ が $\mathcal{A}$ によってコントロールされている場合，$\mathcal{S}$ は $GV$ の *Step 1* におけるゼロ知識シミュレータを走らせプロトコルにおける $U$ の動きをシミュレートする．

**機関登録証明書の発行**

ユーザは $O$ の登録証明書の発行を証明書発行機関 $IO$ に要求する:

(I) ユーザが $\mathcal{A}$ にコントロールされている場合，(i) $T$ からのメッセージを受け取るとすぐに，$\mathcal{S}$ は $x, s_{(U,G)}, s_{(U,O)}$ と $r$ を手に入れるために $CI$ の *Step 1* における知識抽出をおこなう．(i-1) もし $(x, s_{(U,G)}) \notin archive_{IO}$ ならば，$\mathcal{S}$ は証明書の発行を拒否する．(i-2) もし $(x, s_{(U,G)}) \in archive_{IO}$ ならば，$\mathcal{S}$ は $G$ の残りの動きをおこない，$E$ に対応する $c'$ を発行し，$c'/r$ によって $C$ を求める．$\mathcal{S}$ は $archive_O$ に $(x, s_{(U,O)}, C, E)$ を保管する．

(II) もしユーザが $\mathcal{A}$ によってコントロールされており，機関 $O$ が不正直である場合，(i) $T$ からのメッセージを受け取るとすぐに，$\mathcal{S}$ は $CI$ の *Step 1* での知識抽出をおこない $x, s_{(U,G)}, s$ と $r$ を得，$archive_O$ をチェックする．(i-1) もし $(x, s) \in archive_O$ ならば，$\mathcal{S}$ はこのユーザを $U$ とする．(i-2) もし $(x, s) \notin archive_O$ ならば，$U$ を $x$ に対応するユーザとし，$T$ と対話することによって $N_{(U,O)}$ を得る．(ii) $\mathcal{S}$ は $archive_{IO}$ をチェックする．(ii-1) もし $(x, s_{(U,G)}) \notin archive_{IO}$ ならば，$\mathcal{S}$ は証明書の発行を拒否する．(ii-2) もし $(x, s_{(U,G)}) \in archive_{IO}$ ならば $\mathcal{S}$ は $IO$ の残りの作業をおこない，$E$ に対応する正しい $c'$ を作成する．$\mathcal{S}$ は $c'/r$ によって $C$ を求め，$(x, s_{(U,O)}, C, E)$ を $archive_O$ に保管する．

(III) 証明書発行機関 $IO$ が $\mathcal{A}$ によってコントロールされている場合，$\mathcal{S}$ は $CI$ にの *Step 1* をゼロ知識シュミレートし，ユーザ側の残りの行動をおこなう．もし，ユーザが受理するならば，$\mathcal{S}$ は $T$ に証明書が発行されたことを知らせる．

**機関登録証明**

このシミュレーションは，以下の検証機関に対する機関情報をもたない機関属性証明におけるシミュレーションから容易に導くことができる．

**検証機関に対する機関登録証明**

ユーザは検証機関 $O_j$ に $O_i$ 機関への登録証明をおこなう:

(I) ユーザは $\mathcal{A}$ によってコントロールされており，登録証明書を発行した $IO$ は正直であり，検証機関 $O_j$ も正直である場合，(i) $\mathcal{S}$ は $CT$ の $O_j$ 側の動き

をおこない，補題の知識抽出によって$x, s_{(U,O_i)}, s_{(U,O_j)}, E, C$を得る．(i-1) もし$(x, s_{(U,O_i)}, E, C) \notin archive_{O_i}$ならば$\mathcal{S}$は拒否する．(i-2) もし$(x, s_{(U,O_i)}, E, C) \in archive_{O_i}$ならば$\mathcal{S}$は$U$による保証書の保持証明において$T$と通信する．

(II) ユーザは$\mathcal{A}$にコントロールされており，登録証明書を発行した$IO$は不正直であり，検証機関$O_j$が正直である場合，(i) $\mathcal{S}$は$x, s, s_{(U,O_j)}, E, C$を手に入れるため，知識抽出を用いて$CT$における$O_j$の動きをシュミレートする．ここで，$O_i$を公開鍵として$y$をもつ機関とする．(i-1) もし$O_j$側がプロトコルの実行を拒否した場合，$\mathcal{S}$は何もしない．(i-2) そうでない場合: (2-A-a) もし$x \in archive_{O_i}$ならば，このユーザを$U$とする．(2-A-b) もし$x \notin archive_{O_i}$ならば，$U$を$x$に対応するユーザとし$\mathcal{S}$は$T$との対話によって仮名$N_{(U,O_i)}$を手に入れる．(2-B) もし$(E, C) \notin archive_{O_i}$ならば，$\mathcal{S}$は$BIC$を走らせ$U$の記録にその出力を加える．(2-C) $\mathcal{S}$は$U$による証明書の保持証明において$T$と通信する．

(III) もし検証機関$O_j$が$\mathcal{A}$によってコントロールされている場合，$\mathcal{S}$はゼロ知識シミュレータを走らせる．

**定理 7** 強RSA仮定のもと，正しい仮名と証明書の組$(P, C, E)$の保持証明をおこなうことができるならば，この組はプロトコル$PG\langle U, O_i\rangle, CI\langle U, GM\rangle$によって生成されたものである．

*Proof.* $K_A$を攻撃者が要求できる証明書の数，$K_H$を正直なユーザが要求できる証明書の数とし，$K = K_A + K_H$とおく．

$\mathcal{A}$を適応的に$O_i$とプロトコル$PG, IC$を，$IO$とプロトコル$CI$を動かし，$K$個の$(Y_j, (C_j, E_j))$ $(1 \leq j \leq K)$を得ることができる攻撃者とする．$\mathcal{A}$がある正直な検証者$V$または正直な検証機関$O_j$に$1 \leq j \leq K$に対して$\tilde{x} \in \Gamma, \tilde{e} \in \Lambda$であり，かつ$(y^{\tilde{x}}, \tilde{E}) \neq (Y_j, E_j))$をみたす$(\tilde{x}, \tilde{C}, \tilde{E})$の保持証明をおこなうことができると仮定する．ここで，以下の2つの場合を区別する:

**1.** $\forall j$に対して$\gcd(E_j, \tilde{E}) = 1$のとき: もし$\mathcal{A}$がこれらの保持証明に成功するならば，Game 1を用いて強RSA問題を解くことができる．

**2.** $\gcd(E_j, \tilde{E}) = E_j$を満たす$j$が存在するとき: $\mathcal{A}$がこれらの保持証明に成功するならば，Game 2を用いて強RSA問題を解くことができる．

以下に Game 1 と Game 2 を述べる.

**Game 1:** RSA の法 $n$ と $z \in \mathbb{Z}_n^{*2}$ を用意する．ここで，$n$ は素数 $p', q'$ に対する素数 $p = 2p' + 1$ と $q = 2q' + 1$ の積であり $z$ を位数 $p'q'$ をもつ元とする.

**1.** $v_1, \ldots, v_K \in_R ]-2^{\ell_\Delta} + 2^{2\ell_\Gamma}, 2^{\ell_\Delta}[$ と素数 $E_1, \ldots, E_K \in_R \Gamma$ を選択する.

**2.** $y := z^{\prod_\ell E_\ell} \bmod n$ とする.

**3.** $r \in_R \Gamma$ を選択し，$f := y^r \bmod n$ とする.

**4.** 全ての $1 \leq i \leq K$ に対し，$C_i := z^{(v_i+r)\prod_{\ell \neq i} E_\ell}$ を計算する.

**5.** $d, e, g, h \in_R \mathbb{Z}_n^{*2}$ を選択し，$(n, d, e, f, g, h)$ をグループの公開鍵とする.

**6.** 正直なユーザが $GM$ に証明書の発行を求める場合は，$(P_j = {C_j}^{E_j}/f, E_j, C_j)$ を選ぶ.

**7.** $\mathcal{A}$ と以下のように対話をおこなう:

仮名生成プロトコル $PG$: *Step 1*では$\mathcal{A}$からコミットメント $c_1, c_2$ を受け取り，$PK^2$ から ${c_1}^2 = (d^2)^{\tilde{r}_j}(e^2)^{\tilde{r}'_j}$ かつ ${c_2}^2 = (d^2)^{\tilde{x}_j}(e^2)^{\tilde{r}''_j}$ をみたす $\tilde{x}_j, \tilde{r}_j, \tilde{r}'_j, \tilde{r}''_j$ を抽出する．*Step 2* において，$s_j = v_j - \tilde{x}_j r_1 \in \mathbb{Z}$ を計算する．このとき $s_j \in \Delta$ である．次に，$r = s_j + (2^{\ell_\Delta} - 1) - \tilde{r}_j \bmod (2^{\ell_\Delta+1} - 1)$ を計算し，$r$ を $\mathcal{A}$ に送る．残りはプロトコルに沿っておこなう.

証明書発行プロトコル $CI$: $\mathcal{A}$ から仮名 $P$ を受け取ったあと，$P = y^{v_j}$ をみたす $j$ を求める．もしこれをみたす $j$ が存在しない場合は，$IC$ の実行を中止する．そうでない場合は $(C_j, E_j)$ を $\mathcal{A}$ に送る.

属性証明プロトコル $CS$ と $CT$: 検証者 $V$ または検証機関 $O_j$ として実行する．それぞれの *Step 2* における $PK^2$ において，$({\tilde{c}_1}^2/(e^2)^{\tilde{y}})^{\tilde{E}} \equiv (y^2)^{\tilde{x}} f^2 \equiv z^{2(\tilde{x}+r)\prod_\ell E_\ell}$ をみたす $\tilde{x} \in \Gamma, \tilde{E} \in \Lambda$ と $\tilde{y}, \tilde{c}_1$ の抽出をおこなう.

ここで，$\gcd(\tilde{E}, E_j) \neq 1$ となる $E_j (1 \leq j \leq K)$ が存在するならば $V$ または $O_\ell$ としてプロトコルの実行を続ける．そうでない場合は $\hat{E} := 2(\tilde{x} + r)\prod_\ell E_\ell$ とする．このとき，$\gcd(\tilde{E}, E_j) = 1 (1 \leq j \leq K)$ より，$w := \gcd(\tilde{E}, \hat{E}) = \gcd(\tilde{E}, 2(\tilde{x}+r))$ を得る．拡張ユークリッドの互助法を

用いて，$\alpha\tilde{E}+\beta\hat{E}=w$ をみたす $\alpha,\beta\in\mathbb{Z}$ を求め，$u:=z^{\alpha}(\tilde{c_1}/e^{\tilde{y}})^{\beta} \bmod n$ と $E:=\tilde{E}/w$ とおく．このとき，$2|\tilde{x}+r|<2(2^{\ell_\Gamma+1})<2^{\ell_\Gamma},\tilde{E}\in\Lambda$ より，$\tilde{E}>(\hat{x}+r)$ であり $E>1$ を得る．よって，これらは $u^E\equiv z \pmod n$ をみたす強 RSA 問題の解であり，これらを出力して停止する．

**8.** もし $\mathcal{A}$ が停止したならば，Null を返し停止する．

**Game 2:** RSA の法 $n$ と $z\in{\mathbb{Z}_n^*}^2$ を Game 1 と同様に設定する．

**1.** $v_1,\ldots,v_K\in_R]-2^{\ell_\Delta}+2^{2\ell_\Gamma},2^{\ell_\Delta}[$ と素数 $E_1,\ldots,E_K\in_R\Gamma$ を選択する．

**2.** $k\in_R\{1,\ldots,K\}$ を選択し，$y:=z^{\prod_{\ell\neq k}E_\ell} \bmod n$ とする．

**3.** $r\in_R\Gamma$ を選択し，$C_k:=y^r \bmod n, f:={C_k}^{E_k}/y^{v_k} \bmod n$ とする．

**4.** 全ての $1\leq j\leq K$ かつ $j\neq k$ に対し，$C_j:=z^{(v_j+E_kr-v_k)\prod_{\ell\neq k}E_\ell}$ を計算する．

**5.** $d,e,g,h\in_R{\mathbb{Z}_n^*}^2$ を選択し，$(n,d,e,f,g,h)$ をグループの公開鍵とする．

**6.** 正直なユーザが $GM$ に証明書の発行を求める場合は，$(P_j={C_j}^{E_j}/f,E_j,C_j)$ を選ぶ．

**7.** $\mathcal{A}$ と以下のように対話をおこなう:

仮名生成プロトコル $PG$: *Step 1*では$\mathcal{A}$からコミットメント $c_1,c_2$ を受け取り，$PK^2$ から ${c_1}^2=(d^2)^{\tilde{r}_j}(e^2)^{\tilde{r}'_j}$ かつ ${c_2}^2=(d^2)^{\tilde{x}_j}(e^2)^{\tilde{r}''_j}$ をみたす $\tilde{x}_j,\tilde{r}_j,\tilde{r}'_j,\tilde{r}''_j$ を抽出し，$s_j=v_j-\tilde{x}_jr_1\in\mathbb{Z}$ を計算する．このとき $s_j\in\Delta$ である．次に，$r=s_j+(2^{\ell_\Delta}-1)-\tilde{r}_j \bmod (2^{\ell_\Delta+1}-1)$ を計算し，$r$ を $\mathcal{A}$ に送る．残りはプロトコルに沿っておこなう．

証明書発行プロトコル $CI$: $\mathcal{A}$ から仮名 $P$ を受け取ったあと，$P=y^{v_j}$ をみたす $j$ を求める．もしこれをみたす $j$ が存在しない場合は，$IC$ の実行を中止する．そうでない場合は $(C_j,E_j)$ を $\mathcal{A}$ に送る．

登録証明プロトコル $SC$ と $TC$: 検証者 $V$ または検証機関 $O_j$ として実行する．それぞれの *Step 2*における $PK^2$ において，$({\tilde{c_1}}^2/(e^2)^{\tilde{y}})^{\tilde{E}}\equiv(y^2)^{\tilde{x}}f^2\equiv z^{2(\tilde{x}+rE_k-v_k)\prod_{\ell\neq k}E_\ell}$ をみたす $\tilde{x}\in\Gamma,\tilde{s}\in\Delta,\tilde{E}\in\Lambda,\tilde{c}_1$ と $\tilde{y}$ の抽出をおこなう．

ここで，$\gcd(\tilde{E}, E_k) \neq E_k$ ならば $V$ または $O_\ell$ としてプロトコルの実行を続ける．そうでない場合は，ある $t \geq 1$ を用いて $\tilde{E} = tE_k$ とあらわすことができ，もし $\tilde{x} \geq v_k$ ならば $C := (\tilde{c_1}/e^{\tilde{y}})^t/C_k \bmod n$ とし，そうでないならば，$C := C_k/(\tilde{c_1}/f^{\tilde{y}})^t \bmod n$ とおく．このとき，$\hat{E} := 2(\tilde{x}-v_k)\prod_{\ell \neq k} E_\ell$ とおくことで $C^{E_k} \equiv h^{\tilde{x}-v_k} \equiv z^{\hat{E}} \pmod n$ が得られる．拡張ユークリッドの互助法を用いて，$\alpha E_k + \beta \tilde{E} = 1$ をみたす $\alpha, \beta \in \mathbb{Z}$ を求め，$u := z^\alpha C^\beta \bmod n, E := E_k$ とすれば，これらは $u^E \equiv z \pmod n$ をみたす．この $(u, E)$ は強 RSA 問題の解であり，これらを出力して停止する．

**8.** もし $\mathcal{A}$ が停止したならば，Null を返し停止する．

これまでの議論から，以下の補題と定理を得ることができる．

**補題 9** 実際のプロトコルにおいて攻撃者の得る情報は統計的にシミュレーションで得る情報と識別できない．

**定理 8** 強 RSA 問題仮定と Diffie-Hellman 問題仮定，素因数分解問題の困難さの仮定のもと，提案する匿名証明書方式は安全である．

## 5.5 考察

ここで，提案手法の効率性について考察をおこなう．匿名性を強化した証明書方式 [5.3]，匿名複数証明書方式 [5.4] における登録証明プロトコルにおいて，$\mathbb{Z}_n^*$ 上での剰余乗算に必要な計算コストを $M(n)$ とし，剰余べき演算にバイナリ法を用いた場合におけるユーザ，検証者の計算量を表 5.1 にまとめた．

匿名性を強化した証明書方式は，各々のセキュリティポリシーに応じて，機関の登録証明，もしくはグループ内のある機関への登録証明，というように検証者に与える情報をユーザ自身が制御することのできる方式である．この場合，既存方式 [11] における登録証明とほぼ同程度な計算コストによってユーザの匿名性を強化することができたといえる．

また，検証者から複数 $(m > 1)$ の登録証明が求められた場合，既存の方式では，ユーザ，検証者の計算コストが $m$ に比例するが，提案する匿名複数証明書方式では提示する機関の数に依存しない計算量となっている．

表 5.1: 匿名証明書方式における効率性の比較

| | | ユーザの計算量 | 検証者の計算量 |
|---|---|---|---|
| [11] | 登録証明 | $21\|n\|M(n)$ | $370\|n\|/32M(n)$ |
| | $m$ 個の機関への登録証明 | $m \cdot 21\|n\|M(n)$ | $m \cdot 370\|n\|/32M(n)$ |
| [5.3] | 登録証明 | $24\|n\|M(n)$ | $371\|n\|/32M(n)$ |
| | 機関情報をもたない登録証明 | $21\|n\|M(n)$ | $370\|n\|/32M(n)$ |
| [5.4] | $m$ 個の機関への登録証明 | $18\|n\|M(n)$ | $368\|n\|/32M(n)$ |

## 5.6 まとめ

既存の方式 [11] では，ユーザの取引相手となる各機関が秘密鍵，公開鍵 $(n_{O_i}, g_{O_i}, h_{O_i}, f_{O_i} \in {\mathbb{Z}^*_{n_{O_i}}}^2)$ の組を用意し，

$$C_{(U,O_i)}{}^{E_{(U,O_i)}} \equiv {g_{O_i}}^{x_U} {h_{O_i}}^{s_{(U,O_i)}} f_{O_i} \pmod{n_{O_i}}$$

をみたす $C_{(U,O_i)}, E_{(U,O_i)}$ を登録証明書としてユーザに発行していた．但し，$x_U, s_{(U,O_i)}$ はユーザの秘密情報である．

しかし，各機関の特性のみを検証者に示したい場合などは，機関の公開鍵を必要とする登録証明では実現することができず，機関の特性を証明する別機関に登録証明書を発行してもらう必要があった．この場合，それらの証明書を用途に応じて使い分けなければならない．よって，各機関を予め適当なグループに分類し，その機関管理者 $M_G$ が証明書を発行するという方式を提案した．$M_G$ は，グループ $\boldsymbol{G}$ の秘密鍵，公開鍵 $(n_G, g_G, h_G, v_G, f_G \in {\mathbb{Z}^*_{n_G}}^2)$ の組を用意し，機関 $O_i$ の識別情報 $id_{(O_i,G)}$ を用いて，

$$C_{(U,O_i)}{}^{E_{(U,O_i)}} \equiv {g_G}^{x_U} {h_G}^{s_{(U,O_i)}} {v_G}^{id_{(O_i,G)}} f_G \pmod{n_G}$$

をみたす $C_{(U,O_i)}, E_{(U,O_i)}$ を $O_i$ への登録証明書として発行する．これによって，上式をみたす $(C_{(U,O_i)}, E_{(U,O_i)}, x_U, s_{(U,O_i)})$ の知識証明をおこなうことで，機関 $O_i \in \boldsymbol{G}$ への登録を証明でき，加えて $id_{(O_i,G)}$ の知識証明では $O_i$ の情報を検証者に与えないため，グループ内のある機関に登録しているという事実のみを示すことができる．1 つの証明書によってその登録証明スタイルを選択できるということは，ユーザのデータ管理の負担を軽くするとともに，検証者に与える情報をユーザ自身が制御できることから，個人情報の保護を強化した実用化に適する方式であるといえる．

また，複数機関への登録を効率よく証明することのできる手法を，システム内に証明書発行機関 $IO$ を設けることで実現した．$IO$ は，秘密鍵，公開鍵 $(n, g, h, f \in \mathbb{Z}_n^{*2})$ の組を用意し，機関 $O_i$ の公開鍵 $y_{O_i}$ に対して，

$$C_{(U,O_i)}{}^{E_U} \equiv {y_{O_i}}^{x_U} f_G \pmod{n_G}$$

をみたす $C_{(U,O_i)}, E_U$ を $O_i$ の登録証明書としてユーザに発行する．ユーザの全ての属性を $\mathcal{O}_U$ とし，検証者に証明したい機関の集合を $\mathcal{O}_S \subset \mathcal{O}_U$ としたとき，$C_U := \prod_{\mathcal{O}_S} C_{(U,O_i)}$ に対して，

$$C_U{}^{E_U} \equiv (\prod_{\mathcal{O}_S} y_{O_i})^{x_U} f^{|\mathcal{O}_S|} \pmod{n}$$

をみたす $C_U, E_U, x_U$ の知識証明をおこなうことで $\mathcal{O}_S$ の属性を証明することができ，検証者の計算コストは機関数に依存することなく，常に小さく保つことができる．またこの場合，全ての機関の公開鍵 $y_{O_i}$ は，その他の任意の機関の公開鍵の積とならないよう設定しなければならない．

また，これら 2 種類の匿名証明書方式を組み合わせることができる．このとき，匿名性を強化できる登録証明書の複数提示が可能となる．この場合，機関 $O_i$ への匿名性を強化できる登録証明書は，$O_i$ の識別情報 $id_{(O_i,G)}$，証明書発行機関の公開鍵 $g, h, d, f, n$ を用いて

$$C_{(U,O_i)}{}^{E_U} \equiv g^{x_U} h^{s_{(U,O_i)}} d^{id_{(O_i,G)}} f \pmod{n}$$

をみたす $C_{(U,O_i)}$ として与えられるため，$\mathcal{O}_S$ の登録証明書をかけあわせて

$$(\prod_{\mathcal{O}_S} C_{(U,O_i)})^{E_U} \equiv (g^{|\mathcal{O}_S|})^{x_U} h^{\sum_{\mathcal{O}_S} s_{(U,O_i)}} d^{\sum_{\mathcal{O}_S} id_{(O_i,G)}} f^{\mathcal{O}_S} \bmod n$$

をみたす $\prod C_{(U,O_i)}, E_U, x_U, \sum s_{(U,O_i)}$ の知識証明をおこなうとよい．但し，この場合にも全ての機関の識別情報 $id_O$ は他の任意の $id$ の和とならないよう設定する必要がある．

# 第 6章

# 電子オークション

## 6.1 はじめに

現在，最も広く普及しているインターネット上のオークションはイングリッシュ・オークション [42, 43] である．イングリッシュオークションとは，公開される現在の落札価格より高い値を順に入札していき，最終的な最高入札額が落札額，その入札者が落札者となる価格吊り上げ型のオークションである．インターネット上で開催されるイングリッシュ・オークションにおいて商品を落札するためには，参加者は他の入札値を随時チェックして入札を繰り返す必要がある．そのような欠点を克服した方式が代理入札 (Proxy-bidding) である．代理入札とは，各参加者は希望落札額を代理人に提示し，代理人が参加者に代わって入札をおこなうため，落札者決定まで，参加者はインターネットに束縛される必要がない．このような代理入札は次のようにおこなわれる．例えば，開始価格が 1,000 円で，入札幅が 100 円であったとしよう．ここで，参加者 $A$ が落札希望額として 2,000 円を秘密に代理人に提示し，外出したとする．代理人は，現在の落札価格が落札希望額より低いときに限り，入札を続ける．この場合，代理人は 1,000 円を入札し，現在の落札額は 1,000 円，現在の落札者は $A$ となる．次に，別の参加者 $B$ が落札希望額 2,500 円を代理人に提示した場合，$B$ の代理人は公開されている現在の落札価格が 1,000 円なので，それより高い 1,100 円を入札する．この時点で，現在の落札価格は 1,100 円，現在の落札者は $B$ となる．一方，$A$ の代理人はこれに対し 1,200 円を次に入札する．このように代理人による入札は繰り返され，最終的に現在の落札価格は

2,100 円，現在の落札者は $B$ となる．落札者が決定するまで，$A$ が再度，落札希望額を変更することも可能である．

現在のインターネット上でのオークションは，代理人の役割をオークション管理者が一括して担うことで実現されている [2, 1]．この場合，オークション管理者は新たな入札 (落札希望額の提示) がおこなわれたとき，その時点での落札者の落札希望額と新たな入札者の落札希望額を比較し，低額 (+入札幅) を，現在の落札価格とすればよいのである．つまり，上の例では，$B$ の希望落札額である 2,500 円より低い $A$ の希望落札価格 2,000 円に 100 円を加算した額が現在の落札価格となる．このとき，現在の落札者は $B$ となり，次の入札がおこなわれたときには，新たな入札値と 2,500 円を比較することになる．

ここで，代理入札システムの構築に必要となる性質をまとめる．

**入札値の秘匿性** – オークション管理者，参加者等の全てのエンティティに対して，全ての入札値は現在の落札価格の決定 (価格更新) 前まで秘匿され，落札者の入札値は価格更新後も秘匿される．

**匿名性** – オークション管理者，参加者等の全てのエンティティは入札値とその入札者の関係を知ることはできず，落札者以外の入札者は秘匿される．

**公開検証性** – 誰もがオークションの正当性を検証することができる．

**効率性** – 入札，開札における通信量，及び計算量が効率的である．

## 6.2 準備

### 6.2.1 代理人入札

代理人入札に必要となるエンティティ，及びオークションの流れについて述べる．

登録管理者 ($RM$) : オークションの参加者の登録・削除をおこない，参加者の登録情報を知る唯一のエンティティ．

オークション管理者 ($AM_j$) : オークションの入札・価格更新をおこなう $n$ ($\geq 1$) 人のエンティティ．オークション管理者の代表を $AM_1$ とする．

参加者 ($B_i$) : $RM$ へ登録をおこなうことで，オークションの参加資格をもつエンティティ．

代理人入札システムは，以下の 5 つのプロトコルで構築される:

1. 初期設定: $n$ 人のオークション管理者は，暗号化関数とともに公開鍵を生成し，$AM_1$ がそれらを公開する．また，復号関数の秘密鍵は $n$ 人のオークション管理者で分散する．

2. オークション準備: 登録管理者 $RM$ とオークション管理者 $AM_1$ はオークション毎に，各参加者に必要な情報を設定し，それらを公開する．また，オークション管理者 $AM_1$ はそれぞれのオークションに関する，[商品，出品者情報，オークション開始価格，現在の落札価格，オークション終了までの時間] などを公開する．

3. 参加者登録: 参加者は，オークションに参加するため，登録鍵を登録し，オークション参加に必要となる情報を取得する．

4. 入札: 参加者 $B_i$ は，希望落札額 $\boldsymbol{b}_i$ をオークション管理者の公開鍵暗号で暗号化し，入札値とする．

5. 価格更新: $n$ 人のオークション管理者は，新たな入札値と現在の落札者の入札値を暗号化されたまま比較し，低い方の入札値のみを出力する．さらに，価格更新ルールに従い，新たな現在の落札値を更新する．

6. 落札者決定: オークション管理者 $AM_1$ は落札者に関する情報を登録管理者に送り，登録管理者はその情報から落札者を決定する．

代理人入札の価格更新について述べる．新たな入札者を $B_{new}$，$B_{new}$ の落札希望額を $\boldsymbol{b}_{new}$ とする．このとき，オークション管理者は暗号化された入札値 $E(\boldsymbol{b}_{new})$ と，現在の落札者 $B_{high}$ の暗号化された $B_{new}$ による入札値 $E(\boldsymbol{b}_{high})$ を比較し，高い値の入札者を現在の落札者，低い方の値 $(+\Delta)$ を現在の落札価格 $\boldsymbol{b}_{cur}$ とする．ここで $\Delta$ を入札幅とする価格更新に関するルールは以下のようになる．

*1.* $\boldsymbol{b}_{new} < \boldsymbol{b}_{cur} + \Delta$ ならば，$\boldsymbol{b}_{new}$ を拒否.

*2.* $\boldsymbol{b}_{cur} + \Delta \leq \boldsymbol{b}_{new}$ のとき，

*2-1.* $\boldsymbol{b}_{new} < \boldsymbol{b}_{high}$ ならば，$\boldsymbol{b}_{cur} := \boldsymbol{b}_{new} + \Delta, \boldsymbol{b}_{high} := \boldsymbol{b}_{high}, B_{high} := B_{high}$，

*2-2.* $\boldsymbol{b}_{new} = \boldsymbol{b}_{high}$ ならば，$\boldsymbol{b}_{cur} := \boldsymbol{b}_{new}, \boldsymbol{b}_{high} := \boldsymbol{b}_{high}, B_{high} := B_{high}$，

*2-3.* $\boldsymbol{b}_{high} < \boldsymbol{b}_{new}$ ならば，$\boldsymbol{b}_{cur} := \boldsymbol{b}_{high} + \Delta, \boldsymbol{b}_{high} := \boldsymbol{b}_{new}, B_{high} := B_{new}$.

## 6.3 代理人入札

代理人入札システムは，1. 初期設定，2. 準備，3. 参加者登録，4. 入札，5. 価格更新，6. 落札者の決定からなるが，本稿での目的は価格更新に伴う計算・通信コストの削減であり，その他のプロトコルは塩月・宮地 [52] と同様の手続きをとるものとする．本章では，提案する代理人入札システムを準備述べ，価格更新において既存方式と提案方式の 2 つを挙げ，後章にてそれらの効率を比較することにする．

### 1. 初期設定

登録管理者は，参加者登録のための情報として，素数 $p = 2q + 1$ をみたす素数 $p, q$，生成元 $g \in \mathbb{Z}_p^*$ を公開する．オークション管理者 $AM_1$ は，公開鍵暗号の暗号化関数 $E$，及び $u \in \{0,1\}^*$ を選び，それらを公開する．また，各 $AM_i$ は分散情報生成プロトコル [25] を用いて，復号関数 $D$ の秘密鍵の分散情報を作成し，秘密鍵に対応する公開鍵を生成する．

### 2. 準備

オークション毎に，登録管理者 $RM$ は，秘密の乱数 $r_{RM} \in \mathbb{Z}_q$ に対して $y_{RM} = g^{r_{RM}} \bmod p$ を公開する．また，オークション管理者 $AM_1$ も，乱数 $r_{AM} \in \mathbb{Z}_q^*$ に対して $y_{AM} = {y_{RM}}^{r_{AM}} \bmod p$ を公開し，$r_{AM}$ を秘密に保管する．

### 3. 参加者登録

各参加者 $B_i$ は，秘密鍵として $x_i \in_R \mathbb{Z}_q^*$ を選択し，登録鍵として $y_i := g^{x_i} \bmod p$ を $SPK\{(\alpha) : y_i = g^{\alpha}\}(*)$ とともに $RM$ へ送信する．

オークション毎に $RM$ は参加者の登録鍵 $y_i$ に対して，$R_i := {y_i}^{r_{RM}} \bmod p$ を計算し，$RM$ の公開掲示板 $PPT_{RM}$ に登録者リスト $\mathcal{R} = \{R_j\}$ を掲示する．その際，各参加者が他の参加者の登録鍵 $\{y_j\}$ とリスト上の $\{R_j\}$ の対応づけができないよう，$RM$ は全ての登録鍵をシャッフルしたリストを公開するものとする．また，$RM$ は登録鍵 $y_i$ と参加者の $ID_i$ のペアを秘密に保管する．

$AM$ は，$\mathcal{R} \ni R_i$ に対して $T_i := {R_i}^{r_{AM}} \bmod p$ を計算し，オークション管理者の公開掲示板 $PPT_{AM}$ にオークション・チケットリストとして $\mathcal{T} = \{T_j\}$ を掲示する．

### 4. 入札

参加者は $T_i = {y_{AM}}^{x_i} \bmod p$ を計算し，$\mathcal{T}$ 上にオークション・チケットが発行されているかどうかを確認する．$T_i \in \mathcal{T}$ であれば，2 進数であらわした $k$ ビットの希望落札額 $\boldsymbol{b}_i = (b_i^{(k-1)}, \ldots, b_i^{(0)})$ に対して，入札値 $E(\boldsymbol{b}_i) = (e_i^{(k-1)}, \ldots, e_i^{(0)})$ を作成し，$(E(\boldsymbol{b}_i), T_i)$ を公開する．ここで，$E(\boldsymbol{b}_i)$ の各ビットを

$$e_i^{(j)} = \begin{cases} E(u) & b_i^{(j)} = 1 \text{ のとき} \\ E(1) & \text{それ以外} \end{cases}$$

とする．さらに，

$$e_i^{(j)} \in E_1 \cup E_u \quad (0 \leq j \leq k-1)$$

の知識証明によって $E(\boldsymbol{b}_i)$ の正当性を，オークション・チケット・リスト $\mathcal{T} \ni T_i$ に対する $SPK\{(\alpha) : T_i = {y_{AM}}^{\alpha}\}(*)$ によって正しい参加者であることを証明する．

### 5. 価格更新

$n$ 人のオークション管理者は，新たな入札がおこなわれたとき，価格更新ルールに従い，以下の手順で現在の価格 $\boldsymbol{b}_{cur}$，現在の落札者 (オークション・チケット) $T_{high}$，現在の落札者の入札値 $E(\boldsymbol{b}_{high})$ を更新する．

ここで，暗号化された値の大小比較をおこない，低い方の値を出力し，高い方の値に関する情報は何ら与えない関数を

$$Compare(E(\boldsymbol{b}_1), E(\boldsymbol{b}_2)) = \begin{cases} (1, \boldsymbol{b}_1) & \boldsymbol{b}_1 \leq \boldsymbol{b}_2 \text{のとき} \\ (0, \boldsymbol{b}_2) & \text{それ以外} \end{cases}$$

とする．

*Step 1.* オークション管理者 $AM_1$ は，現在の価格 $\boldsymbol{b}_{cur}$ に対し，$E(\boldsymbol{b}_{cur} + \Delta)$ を生成する．

*Step 2.* 新たな入札値 $E(\boldsymbol{b}_{new})$ に対して，

$$Compare(E(\boldsymbol{b}_{cur} + \Delta), E(\boldsymbol{b}_{new})) = (0, \boldsymbol{b}_{new})$$

ならば，入札値 $E(\boldsymbol{b}_{new})$ を拒否．

*Step 3.* $E(\boldsymbol{b}_{new}), E(\boldsymbol{b}_{high})$ に対し，

$$Compare(E(\boldsymbol{b}_{new}), E(\boldsymbol{b}_{high})) = (1, \boldsymbol{b}_{new})$$

ならば，平文等価テストを用いて，$c = E(\boldsymbol{b}_{new})E(\boldsymbol{b}_{high})^{-1}$ に対し，

$$(\boldsymbol{b}_{cur}, T_{high}, E(\boldsymbol{b}_{high})) := \begin{cases} (\boldsymbol{b}_{new}, T_{high}, E(\boldsymbol{b}_{high})) & c \in E_1 \text{のとき} \\ (\boldsymbol{b}_{new} + \Delta, T_{high}, E(\boldsymbol{b}_{high})) & \text{それ以外} \end{cases}$$

とする．

そうでないならば，

$$(\boldsymbol{b}_{cur}, T_{high}, E(\boldsymbol{b}_{high})) := (\boldsymbol{b}_{high} + \Delta, T_{new}, E(\boldsymbol{b}_{new}))$$

とする．

ここで，大小比較関数 $Compare$ として金持ちの財産比べプロトコルを用いた既存方式と，ビット・スライス・プロトコルを用いた提案方式をそれぞれ与える．但し，各関数への入力は $k$ ビットの 2 進数 $\boldsymbol{b}_i = (b_i^{(k-1)}, \ldots, b_i^{(0)})$ に対して，

$$e_i^{(j)} = \begin{cases} E(u) & b_i^{(j)} = 1 \text{ のとき} \\ E(1) & \text{それ以外} \end{cases}$$

としたときの $E(\boldsymbol{b}_i) = (e_i^{(k-1)}, \ldots, e_i^{(0)})$ とする．

**既存方式での価格更新** [52]:

*Step 1.* $a_k = E(u)$ とし，ミックス・アンド・マッチにより，$k-1 \geq j \geq 1$ に対し，

$$s_j := T_{(eq)}(e_1^{(j)} e_2^{(j)}), \quad a_j := T_{(and)}(a_{j+1} s_j)$$

を求める．

但し，$T_{(eq)}$ は左列 $e \in \{E(1), E(u), E(u^2)\}$ に対し，

$$s := \begin{cases} E(1) & e \in E_u \text{のとき} \\ E(u) & \text{それ以外} \end{cases}$$

を右列にもつテーブルであり，$T_{(and)}$ は

$$a := \begin{cases} E(u) & e \in E_{u^2} \text{のとき} \\ E(1) & \text{それ以外} \end{cases}$$

を右列にもつテーブルである．

*Step 2.* $k-1 \geq j \geq 0$ に対し，

$$t_j := T_{(big)}(e_1^{(j)} e_2^{(j)}), \quad u_j := T_{(and)}(a_{j+1} t_j)$$

を求める．

但し，$T_{(big)}$ は左列 $e \in \{E(1), E(u), E(u^{-1})\}$ に対し，

$$t := \begin{cases} E(u) & e \in E_u \text{のとき} \\ E(1) & \text{それ以外} \end{cases}$$

を右列にもつテーブルである．

*Step 3.* $v_0 := u_0$ とし，$1 \leq j \leq k-1$ に対し，

$$v_j := T_{(or)}(v_{i-1} u_j)$$

を求める．$T_{(or)}$ は左列 $e \in \{E(1), E(u), E(u^2)\}$ に対し，

$$v := \begin{cases} E(1) & e \in E_1 \text{のとき} \\ E(u) & \text{それ以外} \end{cases}$$

を右列にもつテーブルである．

*Step 4.* $v_{k-1}$ を閾値分散復号し，$D(v_{k-1}) = u$ ならば $E(\boldsymbol{b}_1)$ を，$D(v_{k-1}) = 1$ ならば $E(\boldsymbol{b}_2)$ を閾値分散復号し，

$$(c, \boldsymbol{b}_{low}) := \begin{cases} (1, \boldsymbol{b}_1) & D(v_{k-1}) = u \text{ のとき} \\ (0, \boldsymbol{b}_2) & \text{それ以外} \end{cases}$$

を $Compare(E(\boldsymbol{b}_1), E(\boldsymbol{b}_2))$ の結果として出力する．

**提案する価格更新**:

*Step 1.* $\boldsymbol{w} = (E(1), E(1))$ とし，$j = k - 1$ とし，$j = 0$ まで繰り返す;

*Step 1-1.* $\boldsymbol{w} = (w_1, w_2)$ において，ミックス・アンド・マッチを用いて，

$$s_1^{(j)} := T_{(or)}(w_1 e_1^{(j)}), s_2^{(j)} := T_{(or)}(w_2 e_2^{(j)})$$

を求め，$\boldsymbol{s}_j := (s_1^{(j)}, s_2^{(j)})$ とする．

*Step 1-2.* $h_j = s_1^{(j)} s_2^{(j)}$ とし，平文等価テストを用いて，

$$b^{(j)} = \begin{cases} 1 & h_j \in E_{u^2} \text{ のとき} \\ 0 & \text{それ以外} \end{cases}$$

とする．

*Step 1-3.* $b^{(j)}$ の結果を受けて，$\boldsymbol{w}$ に

$$\boldsymbol{w} = \begin{cases} \boldsymbol{w} & b^{(j)} = 1 \text{ のとき} \\ \boldsymbol{s}_j & \text{それ以外} \end{cases}$$

を代入し，$j = j - 1$ とし，*Step 1-1* へ．

*Step 2.* $\boldsymbol{w}$ を閾値分散復号し，$\boldsymbol{b}_{low} = (b^{(k-1)}, \ldots, b^{(0)})$ に対して

$$(c, \boldsymbol{b}_{low}) := \begin{cases} (1, \boldsymbol{b}_{low}) & D(\boldsymbol{w}) = (1, u) \text{ のとき} \\ (0, \boldsymbol{b}_{low}) & \text{それ以外} \end{cases}$$

を $Compare(E(\boldsymbol{b}_1), E(\boldsymbol{b}_2))$ の結果として出力する．

**6. 落札者決定**

オークション管理者 $AM_1$ は，オークション終了時の $\boldsymbol{b}_{cur}, T_{high}$ をそれぞれ落札値，落札者のオークション・チケットとし，$R' := {T_{high}}^{1/r_{AM}} \bmod p$ と共に公開し，$\mathcal{R} \ni R'$ に対する $SPK\{(\alpha) : T_h = {R'}^{\alpha}\}(*)$ とともに公開する．

登録管理者は $y' := {R'}^{1/r_{RM}} \bmod p$ を計算し，$SPK\{(\alpha) : R' = {y'}^{\alpha}\}(*)$ とともに $y' = y_j$ をみたす参加者を落札者として公開する．

## 6.4 考察

### 6.4.1 安全性

本章では，提案する代理人入札の安全性について考察する．

**入札値の秘匿性** – 各参加者が提示する入札値は，オークションシステムの暗号化関数を用いて暗号化されており，秘密鍵は $n$ 人のオークション管理者に分散されているため，価格更新前に $t$ $(\leq n)$ 人未満のオークション管理者によって復号することは不可能である．また，価格更新で用いる関数 $Compare$ は，低い方の値を出力するが，高い方の値に関する情報は何も漏らさない．

**匿名性** – 登録者のデータ (登録鍵，$ID$) は，登録管理者によって管理されている．オークション中はオークション・チケットを用いて入札が行われるが，それらはオークション管理者によってランダム化されているため，登録管理者はオークション管理者と結託しない限り入札者と入札値の対応を知ることができない．

**公開検証性** – 入札においては，入札値 $E(\boldsymbol{b}_i) = (e_i^{(k-1)}, \ldots, e_i^{(0)})$ の正当性は $e_i^{(j)} \in E_1 \cup E_u$ $(0 \leq j \leq k-1)$ の知識証明によって，またオークション・チケット $T_i$ に対する $SPK\{(\alpha) : T_i = {y_{AM}}^{\alpha}\}$ の検証によって参加者の正当性を確認できる．

価格更新においては，暗号化された入札値を公開とすることによって，ミックス・アンド・マッチの公開検証性より，誰もが正しく価格更新がおこなわれたことを確認することができる．

落札者決定においては，落札者のオークション・チケット $T_{high}$ に対して，オークション管理者 $AM_1$ によって生成された $SPK\{(\alpha) : T_{high} = R'^{\alpha}\}$ と，登録管理者 $RM$ による $SPK\{(\alpha) : R' = y'^{\alpha}\}$ によって，確かに $y'$ に対応する参加者が落札者であることが検証できる．

**効率性** – 入札，開札における計算コストについては，次章にて後述する．

### 6.4.2 効率性

ここで，提案方式の計算量について考察する．入札において，各参加者は $k$ 回の暗号化と $k$ 回の知識証明をおこなう．また，価格更新においては，手法 1 を用いた場合，1 回の $Compare(E(\boldsymbol{b}_1), E(\boldsymbol{b}_2))$ の計算に $5k-3$ 回のミックス・アンド・マッチと *Step 4* での閾値分散復号を必要とし，手法 2 を用いた場合は，1 回の $Compare(E(\boldsymbol{b}_1), E(\boldsymbol{b}_2))$ の計算に $2k$ 回のミックスアンドマッチと $k$ 回の PET，そして *Step 2* での閾値分散復号が必要となる．

ここで，テーブル $T_{(eq)}, T_{(and)}, T_{(or)}, T_{(big)}$ を用いた 1 回のミックス・アンド・マッチには，それぞれ 1 回の暗号文乗算と 3 回の PET が必要となるため，既存方式 [52] には $15k-9$ 回の PET と 1 回の閾値分散復号，提案手法では $7k$ 回の PET と 2 回の閾値分散復号が必要となる．ここで，[52] と提案手法を比べると，演算に用いる群の位数 $q$ を 160 ビット，オークション管理者への秘密分散閾値を $t=2$ とし，剰余べき演算にバイナリ法を用いた場合，1 回の PET に必要な剰余乗算の回数は $29|q| = 4640$ 回であり，1 回の閾値分散復号に必要な剰余乗算の回数が $39|q|/4 = 1560$ であることから，これは約 1/3PET であることから，これらの比較は PET の回数のみでおこなうことができ，提案手法のほうが効率がよいといえる．

また，関数 $Compare$ の構築には第二価格入札を用いることができる．阿部・鈴木による第二価格入札 [5] を適用した場合と，本提案方式を比較する．この手法では，参加者はオークション管理者によって与えられた離散値リスト $List = \{price_\ell, \ldots, price_1\}$ から入札値を選択するため，$k$ ビットの整数を表現するには $2^k$ ビットの入札値が必要である．よって，入札には，$2^k$ 回の暗号化と $2^k-1$ 回のゼロ知識証明，価格更新における 1 回の $Compare$ の計算には $2^{k+2}-2$ 回の剰余乗算と $k$ 回の PET が必要となる．1 回の PET に必要となる剰余乗算の回数は 4640 で

あることから，1 回の剰余乗算を約 $2^{-12}$PET と見積もると，$Compare$ の計算には $2^{k-10}+k$ 回の PET が必要であるということができる．このとき，$k$ が 7 以上である場合，提案方式のほうが効率がよいといえる．

これらを表 6.1 に，それぞれ暗号化，知識証明，PET の回数によって比較した結果をまとめる．

表 6.1: 代理入札方式における効率性の比較

| | | [52] | 提案方式 | [5] |
|---|---|---|---|---|
| 入札 | 暗号化 | $k$ | $k$ | $2^k$ |
| | 知識証明 | $k$ | $k$ | $2^k+1$ |
| 価格更新 | PET | $15k-9$ | $7k$ | $2^{k-10}+k$ |

## 6.5 まとめ

代理人入札システムは現在最も広く普及している電子オークション・スタイルである．現在使用されている代理人入札システム [2, 1] は，オークション管理者に対する入札値の秘匿性や匿名性をもたないため，ユーザにとって真に安全なシステムとはいえない．本章で提案したシステムは，電子オークションのみたすべき性質を全て兼ね備えており，価格更新時に必要な計算コストは第二価格入札 [5] の適用，及び既存の代理人入札 [52] より低く抑えることができるため，リアルタイム処理に強く実用化に適した手法であるといえる．

# 第 7章

# むすび

インターネットの進展に伴い，多種多様な電子商取引が普及するなか，悪意あるユーザによる犯罪が横行している．そのなかで，個人ユーザはどのように見知らぬユーザ・機関と関わっていくべきか考えなければならない．それは取引の形態によって異なり，個人情報を取引ユーザに与えなければならない場合には，その取引内容，取引ユーザの信頼性を確保しておく必要があり，オンライン・サービスなどを受ける場合などは，自分の属性のみを証明し，その他の情報は何ら漏らさない方式を利用するとよい．

まず，電子商取引にまつわる犯罪を未然に防ぐ手段として，取引に必要となる情報に予め信頼できる第三者の保証をつけるという電子保証の概念を提案した．提案する電子保証方式では，保証者の役割はユーザの役割を包含するという性質を電子的に実現している．これは署名検証における保証書単独検証可能性，署名単独検証不可能性であり，だれが保証者となっている取引なのかということのみを検証することは可能であるが，だれが取引ユーザであるかということのみを検証することはできないという性質を意味する．また，このような性質は署名の長期的安全性の確保に利用することができる．保証者が高い安全性をもつ鍵を利用した場合，計算機のコスト・パフォーマンスの向上等によって，ユーザが用いた鍵の安全性が危惧された状況においても，電子保証のはたらきによって取引内容の正当性を保証することができる．

また，情報の漏洩・拡大を防ぐ手法として匿名証明書方式の提案をおこなった．匿名性を強化した証明書方式は，各機関を適当なグループに分類し，各グループ内

に登録証明書の発行をおこなう機関管理者を用意することによって，登録証明の際に検証者に与える情報をユーザ自らが制御することができる手法である．ユーザは，検証者のセキュリティポリシーに応じて，当該機関に登録している事実の証明，もしくは機関を束ねるグループに登録している事実のみの証明のいづれかの登録証明をおこなうことができる．これにより，ユーザは自らの個人情報をフレキシブルに管理・制御することができる．

また，相違なる複数の機関の登録証明に必要となる計算量を削減するため，ユーザが登録する全ての機関のうち，任意の複数機関の登録を効率よく示すことのできる複数匿名証明書方式を提案した．提案方式における $m$ 個の登録証明に必要となる計算量は既存方式のおよそ $1/m$ で実現可能である．このように複数の登録証明を 1 度の対話によって可能にしたことは，匿名証明書方式の実用化に向けた大きな前進であるといえる．

さらに本稿では，広く普及している電子商取引の一つである代理入札方式にセキュリティ技術を導入することによって，複数のユーザが同時に参加するような取引形態の安全でかつ効率のよいシステムの構築をおこなった．代理入札は，価格更新時に現在の落札者の希望落札価格と入札値を比較し，現在の落札価格を低い方の値によって決定するシステムであり，求められる性質は価格更新前までの入札値の秘匿性と，価格更新後の高い方の値の秘匿性である．また，新たな入札がおこなわれる度に価格更新をおこなう必要があるため，この処理にかかるコストの削減がシステムの効率性に大きく関わってくる．本稿ではビットスライス回路を用いてこれを設計し，既存システムの約 1/2 の計算コストで安全なシステムを実現した．この代理入札はユーザとオークション・システム間の安全性，効率性を兼ね備える方式である．しかしながら，落札者決定後は取引が出品者と落札者間に移行する．よって，落札者決定後の取引に電子保証人方式を用いることで，真の意味での安全な電子商取引が実現できるといえる．

本稿では，高度情報化・ネットワークの進展に伴うインターネットの裾野の広がりにおける電子商取引にまつわる問題について議論してきた．電子商取引においては，個人ユーザがどのように自分自身の個人情報を制御・管理するかということが最も重要な課題である．ユーザは自らの発信する情報をいかにして信頼してもらうか，相手の情報をいかにして信頼するか，また，取引相手の必要とする情

報のみをいかにして効率よく示すことができるかということについて議論し，それらを解決する手段を本稿にて提案した．これらの活用によって，個人ユーザは自らの情報を制御しつつ，安全な電子商取引をおこなうことができるだろう．

# 謝辞

本研究を行なうに当たり, 終始御指導を賜わった宮地 充子助教授に深謝致します.

また, 日頃から有益な御助言をいただき, 多面に渡って励ましていただいた双紙正和特任助教授に感謝致します.

本学の金子 峰雄教授，浅野 哲夫教授，平石 邦彦教授，そして松下電器産業の館林 誠様，千葉大学総合メディア基盤センターの多田 充助教授には，数々の有益な助言をいただきましたことを，心より感謝致します.

さらに, 本論文をまとめるに当たって御協力いただいた宮地研究室の諸兄に厚く御礼申し上げます.

最後に，北陸先端科学技術大学院大学において研究をおこなう機会を与えてくれました両親に心から感謝とお礼を申し上げます.

# 参考文献


[1] `http://auctions.yahoo.co.jp`.

[2] `http://www.ebay.com`.

[3] M. Abe. "Mix-Networks on Permutation Networks". In *Proceedings of Asiacrypt'99*, volume LCNS1716, pages 258–273. Springer-Verlag, 1999.

[4] M. Abe and F.Hoshino. "Remarks on Mix-Network based on Permutation Networks". In *Proceedings of PKC2001*, pages 317–324, 2001.

[5] M. Abe and K.Suzuki. "$M+1$-st Price Auction Using Homomorphic Encryption". In *Proceedings of PKC2002*, pages 115–124, 2002.

[6] G. Ateniese, J.Camenisch, M.Joye, and G.Tsudik. "A practical and provably secure coalition-resistant group signature scheme". In *Proceedings of Crypto 2000*, volume 1880, pages 255–270. Springer Verlag, 2000.

[7] I. B.Damgard. "Payment systems and credential mechanism with provable security against abuse by individuals". In *Proceedings of Crypto'88*, volume 403, pages 328–335. Springer Verlag, 1990.

[8] M. Bellare, C.Namprempre, D.Pointcheval, and M.Semanko. "The Power of RSA Inversion Oracles and the Security of Chaum's RSA-Based Blind Signature Scheme". In *Proceedings of Financial Cryptography, FC'2001*, volume LNCS 2339, pages 319–338. Springer Verlag, 2001.

[9] M. Bellare and P.Rogaway. "The Exact Security of Digital Signatures – How to Sign with RSA and Rabin". In *Proceedings of Eurocrypto'96*, volume

LNCS1070, pages 399–416. Springer-Verlag, 1996.

[10] D. Brown and D. Johnson. "Formal security Proofs for a Signature Scheme with Partial Message Recovery". In *Technical Report CORR 2000-39, Dept. of C & O, University of Waterloo, 2000*, 2000.

[11] J. Camenisch and A.Lysyanskaya. "Efficient non-transferable anonymous multi-show credential system with optional anonymity revocation". In *Proceedings of Eurocrypt 2001*, volume 2045, pages 93–118. Springer Verlag, 2001.

[12] J. Camenisch and A.Lysyanskaya. "Dynamic accumulators and application to efficient revocation of anonymous credentials". In *Proceedings of Crypto 2002*, volume 2442, pages 61–76. Springer Verlag, 2002.

[13] J. Camenisch and E.V.Herreweghen. "Design and implementation of the idemix anonymous credential system". In *ACM CCS'02*, 2002.

[14] J. Camenisch and M.Stadler. "Efficient group signature schemes for large groups". In *Proceedings of Crypto'97*, volume 1294, pages 410–424.

[15] D. Chaum. "Security without identification: Transaction systems to make big brother obsolete". In *Communications of the ACM*, volume 28, pages 1030–1044, 1985.

[16] D. Chaum. "Designated Confirmer Signatures". In *Proceedings of Eurocrypt '94*, pages 86–91, 1994.

[17] D. Chaum and J.-H.Evertse. "A secure and privacy - protecting protocol for transmitting personal informaion between organizations". In *Proceedings of Crypto'86*, volume 263, pages 118–167. Springer Verlag, 1987.

[18] L. Chen. "Access with pseudonyms". In *Cryptography: Policy and Algorithms*, volume 1029, pages 232–243. Springer Verlag, 1995.

[19] R. Cramer and V. Shoup. "'A Practical Public Key Cryptosystem Provably Secure against Adaptive Chosen Ciphertext Attack". In *Proceedings of Crypto'98*, volume LNCS 1642, pages 13–25. Springer-Verlag, 1998.

[20] R. Cramer and V.Shoup. "Signature schemes based on the strong RSA assumption". In *6th ACM Conference on Computer and Communications Security*, pages 46–52. ACM press, 1999.

[21] I. Damgard. "Efficient Concurrent Zero-knowledge in the Auxiliary String Model". In *Proceedings of Eurocrypto 2000*, volume LNCS 1807, pages 431–444. Springer-Verlag, 2000.

[22] T. ElGamal. "A Public-key Cryptosystem and a Signature Scheme based on Discrete Lofarithms". In *IEEE Transactions on Information Theory*, pages 469–472, 1985.

[23] A. Fiat and A.Shamir. "How to prove yourself: Practical solution to identification and signature problems". In *Proceedings of Crypto'86*, volume 263, pages 186–194. Springer Verlag, 1987.

[24] E. Fujisaki and T.Okamoto. "Statistical zero knowledge protocols to prove modular polynomial relations". In *Proceedings of Crypto'97*, volume 1294, pages 16–30. Springer Verlag, 1997.

[25] R. Gennaro, S.Jarecki, H.Krawczyk, and T.Rabin. "Secure Distributed Key Generation for Discrete-Log Based Cryptosystems". In *Proceedings of Eurocrypt'99*, volume 1592, pages 295–310, 1999.

[26] M. Harkavy, D.Tyger, and H.Kikuchi. "Electronic Auctions with Private Bids". In *Proceedings of Symposium on Cryptography and Information Security*, 1998.

[27] M. Jakobsson and A.Juels. "Millimix: Mixing in Small Batches". In *CIMACS Tchnical report 99-33*, 1999.

[28] M. Jakobsson and A.Juels. "Mix and Match: Secure Function Evaluation via Ciphertexts". In *Proceedings of Asiacrypt 2000*, pages 162–177, 2000.

[29] H. Kikuchi. "$(M+1)$st-Price Auction Protocol". In *Proceedings of FC2001*, 2001.

[30] H. Kikuchi, M.Harkavy, and D. Tyger. "Multi-Round Anonymous Auction Protocols". In *Proceedings of the First IEEE Workshop on Dependable and Real-Time E-Commerce Systems*, pages 62–69, 1998.

[31] K. Kurosawa and W.Ogata. "Bit-Slice Auction Circuit". In *Proceedings of ESORICS2002*, volume 2502, pages 24–38, 2002.

[32] A. Lysyanskaya, R.Rivest, A.Sahai, and S.Wolf. "Pseudonym Systems". In *Selected Areas in Cryptography*, volume 1758. Springer Verlag, 1999.

[33] H. Doi M. Mambo E. Okamoto M. Tada Y. Yoshifuji M. Burmester, Y. Desmedt. "A Structured ElGamal-Type Multisignature Scheme". In *Proceedings of PKC'2000*, pages 466–482, 2000.

[34] M. Michels and M. Stadler. "Generic constructions for Secure and Efficient Confirmer Signature Schemes". In *Proceedings of Eurocrypt '98*, pages 406–421, 1998.

[35] S. Mitomi and A. Miyaji. "A multisignature scheme with message flexibility, order flexibility and order verifiability". In *Proceedings of Information security and privacy-Proceedings of ACISP 2000*, pages 298–312, 2000.

[36] M. Naor. "Bit Commitment Using Pseudo-Randomness". In *Proceedings of Crypto '89*, pages 128–136, 1990.

[37] M. Naor, B.Pinkas, and R.Sumner. "Privacy Preserving Auctions and Mechanism Design". In *Proceedings of ACM Conference on Electronic Commerce*, pages 120–127, 1999.

[38] K. Nyberg and R. A. Rueppel. "Message Recovery for Signature Schemes Based on the Discrete Logarithm Problem". In *Designs Codes and Cryptography, 7*, pages 61–81, 1996.

[39] K. Ohta and T. Okamoto. "Multi-signature schemes secure against active insider attacks". In *IEICE transactions of fundamentals, vol. 82-A, no.1*, pages 22–31, 1999.

[40] T. Okamoto. "Provably Secure and Practical Identification Schemes and Corresponding Signature Schemes". In *Proceedings of Crypto'92*, volume LNCS 740, pages 31–53. Springer-Verlag, 1993.

[41] T. Okamoto. "Designated Confirmer Signatures and Public-Key Encryption are Equivalent". In *Proceedings of Crypto '94*, pages 61–74, 1994.

[42] K. Omote and A.Miyaji. "A Practical English Auction with One-time Registration". In *Proceedings of ACISP2001*, volume 2119, pages 221–234. Springer-Verlag, 2001.

[43] K. Omote and A.Miyaji. "A Practical English Auction with Simple Revocation". In *IEICE Trans. Funcamentals*, volume E85-A, No.5, pages 1054–1061, 2002.

[44] P. Paillier. "Public-Key Cryptosystems Based on Composite Degree Residuosity Classes". In *Proceedings of Eurocrypt'99*, volume 1592, pages 223–238, 1999.

[45] L. A. Pintsov and S. A. Vanstone. "Postal Revenue Collection in the Digital Age". In *Proceedings of Financial Cryptography, FC2000*, 2000.

[46] D. Pointcheval and J. Stern. "Security proofs for signature". In *Proceedings of Eurocrypt '96*, pages 387–397, 1996.

[47] C. P.Schnorr. "Efficient signature generation for smart cards". In *Journal of Cryptology*, volume 4, pages 239–252, 1991.

[48] R. Rivest, A. Shamir, and L. Adleman. "A Method for Obtaining Digital Signature and Public-key Cryptosystems". In *Communications of the ACM*, volume 21(2), pages 120–126, 1978.

[49] K. Sako. "Universally Verifiable Auction Protocol which Hides Losing Bids". In *Proceedings of PKC2000*, pages 35–39, 2000.

[50] A. Shamir. "How to Share a Secret". In *Communications of the ACM, 22(11)*, pages 612–613, 1979.

[51] A. Shimbo. "Multisignature Schemes Based on the Elgamal Scheme". In *The 1994 Symposium on Cryptography and Information Security, SCIS94-2C*, 1994.

[52] T. Shiotsuki and A.Miyaji. "An English Auction Protocol with Proxy Bidding". In *Technical Report of IEICE. ISEC2002-37*, pages 25–29, 2002.

[53] A. Yao. "Protocols for Secure Computations (extended abstract)". In *Proceedings of FOCS'82*, pages 160–164. IEEE Computer Society, 1982.

# 本研究に関する発表論文


[1] Yuko TAMURA and Atsuko MIYAJI, “Interactive Signature Scheme between Confirmer and Signer”, IEICE Japan Tech. Rep., ISEC2000-137, pp. 63-70, 2001.

[2] Yuko TAMURA and Atsuko MIYAJI, “A Signature Scheme with a Guarantee”, IEICE Technical Rep., ISEC2001-70, pp. 1-8, 2001.

[3] Yuko TAMURA and Atsuko MIYAJI, “A Signature Scheme with a Guarantee”, 2002 International Symposium on Information Theory and Applications, Proceedings of ISITA2002, S6-6-5, pp763-756.

[4] Yuko TAMURA and Atsuko MIYAJI, “Anonymity-enhanced Pseudonym System’, Proceedings of the 2003 Symposium on Cryptography and Information Security, SCIS2003-3C-3, pp. 149-154, vol. I of II, 2003.

[5] Yuko TAMURA and Atsuko MIYAJI, “Anoymity-enhanced Pseudonym System”, International Conference on Applied Cryptography and Network Security 2003, Lecture Notes in Computer Science, vol. 2846, Springer-Verlag, pp. 33-47, 2003.

[6] 田村 裕子, 宮地 充子, “効率のよい代理人入札システム”, IEICE Japan Tech. Rep., ISEC2003-54, pp. 29-34, 2003.

[7] 田村 裕子, 塩月 徹, 宮地 充子, “効率のよい代理入札システム”, 電子情報通信学会論文誌　マルチメディア社会における情報セキュリティ 採録決定済.

[8] 田村 裕子, 宮地 充子, “複数属性認証システム”, IEICE Japan Tech. Rep., ISEC2003-69, pp. 23-28, 2003.

[9] 田村 裕子，宮地 充子, “実用化に適した匿名証明書方式”，情報処理学会論文誌 プライバシを保護するコンピュータセキュリティ技術特集号 条件付採録決定済.