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

2006－CSEC－34 (17)
2006／7／20

対話証明のバッチ実行

山本 剛† 千田 浩司†

† 日本電信電話株式会社
神奈川県横須賀市光の丘 1-1

あらまし 証明者, 検証者ともに同一のアルゴリズムからなる対話証明について, 多数の実体を実行する場合を考える. 本稿では多数の対話証明を同時に処理するためのプロトコルを構成する方法を提案する. 提案方法はある種の準同型性を持つNP関係について, 多様なHonest Verifierゼロ知識証明プロトコルに適用することができる. また, そのような準同型性を持たない場合についてのバッチ実行について例を挙げて論じる.

キーワード 対話証明, バッチ処理

Batch Processing of Interactive Proofs

Go YAMAMOTO† and Koji CHIDA†

† NTT Corporation
1-1 Hikarinooka, Yokosuka, Japan

Abstract Suppose one have to process many instances of a single interactive proof. We propose a design principle to configure a protocol that executes many instances. Our principle applies various honest verifier ZK protocols for NP-relations with certain homomorphicity. In addition we also propose an actual protocol for an NP-relation without such a homomorphisity.

Key words Interactive Proof, Batch Processing

1. はじめに

対話証明は証明者と検証者の間で対話的に実行される計算である. 通常, 証明者と検証者は確率的なアルゴリズムでモデル化される. なかでもゼロ知識性や知識健全性を持つように構成された対話証明は, ディジタル署名や各種のマルチパーティープロトコルの構成要素として広く応用されている.

実際の応用においては, 同一のアルゴリズムが繰り返し大量に実行されることが想定されることが珍しくない. たとえば, ディジタル署名であれば, 権威を持った単一の検証者が大量の署名を検証するような用途を考えることができるであろう. したがって, 証明者, 検証者ともに同一のアルゴリズムからなる対話証明について, 多数の実体を実行する場合に効率を向上させることは重要な課題である.

同一の対話証明をたとえば n 回にわたって計算することが要求される場合, 証明者や検証者が計算に費やす資源や証明者と検証者の間の通信量は, どれほどが必要であろうか. あきらかに, アルゴリズムを n 繰り返せば必要な資源も n 倍であるが, それらをまとめて実行した場合に我々はもう少し効率のよい方法を考えることができる.

このアイデアは古くからある. たとえば, 相異なる底 $g_1, g_2, .., g_t$ と指数 $x_1, x_2, \cdots x_t$ を与えたときに, $g_1^{x_1} g_2^{x_2} \cdots g_t^{x_t}$ の計算は, $g_i^{x_i}$ を独立して計算するよりも効率よく行うことができる. これをSchnorr署名のようなデジタル署名方式に適用すれば, 検証者の計算量を一定の割合で削減することができる. たとえば [1] にはより効率の良い検証者アルゴリズムが提案されている. 証明者のバッチ実行については, たとえば [7] にあるようなバッチ計算アルゴリズムを用いることができる. また, [2] にはSchnorrのゼロ知識証明プロトコルをバッチ的に実行することで計算量のみならず通信量も削減する方式が記述されている.

本稿では, ある種の準同型性を持つNP関係について対話証明 π が知識健全性を持つときに, その対話証明を計算量・通信量両方の観点で効率よくバッチ実行する方式を定式化し, 安全性を証明する. 提案方式は [2] の方式を例として含むが, もとの対話証明をブラックボックス的に用いて証明するので, 証明は内部の構成に依存しない. したがって提案方式を応用して多様なプロトコルをバッチ実行することができる. 本稿ではそのような例の中から, とくに実用的と思われる例を挙げる.

本稿で扱う準同型性をもつNP関係は多くの例を持つが, 実用的に重要であるにもかかわらず, かならずしもそうではない場合もある. たとえば, Cramerらによる Partial Proof のよ

うに, 知識の間の演算が本質的に困難な場合がある. 本稿では Partial Proof をバッチ実行する方法も論じる. インスタンスがある種の性質を満たすように生成されている場合に有効なバッチ実行プロトコルを提案する.

2. Schnorr 認証のバッチ実行

次の認証プロトコルは [9] で議論されて以来良く知られている. g をある群の元とし, 位数が十分大きい素数 $p = |g|$ であるとしよう.

プロトコル 1.
P の秘密入力: $x \in \mathbb{Z}/p\mathbb{Z}$,
共通入力: $y = g^x$
(1) P は $r \leftarrow \{0,..,p-1\}$ をランダムに選び, $R = g^r$ を計算する. R を V に送信する.
(2) V は $c \leftarrow \{0,..,p-1\}$ をランダムに選び, P に送信する.
(3) P は $z = r - xc \mod p$ を計算し, V に送信する.
(4) V は $g^z = y^{-c}R$ であるか否かを検証し, 成功した場合に証明を受理する.

このプロトコルが [6] などに定義される意味で Honest Verifier 完全ゼロ知識証明であること, 知識健全であることは良く知られている.

プロトコル 1 を認証や署名, あるいはマルチパーティープロトコルの構成要素として用いるとき, 場合によっては同時に大量に実行する必要がある. たとえば d 人の証明者があり, 検証者が一人であるような場合, 検証者の計算資源がボトルネックとなる. そのような場合, [4] にあるような技術を用いることで, 検証者が単一の検証アルゴリズムを d 回繰り返す場合と比較して, 必要とされる計算資源を低減することができる. このアプローチをバッチ検証, あるいはバッチ計算とよぼう.

さて, 一人の証明者と一人の検証者があり, その間でプロトコル 1 を d 回繰り返す必要がある状況ではどうだろうか. この場合, 証明者や検証者の計算資源はもちろんであるが, 証明者と検証者の間の通信量もボトルネックとなり得る. ここでは, [2] で議論されているように, 次に示すような方式を考えることができる.

プロトコル 2.
P の秘密入力: $x_i \in \mathbb{Z}/p\mathbb{Z}(i = 1, 2, \cdots, d)$,
共通入力: $y_i = g^{x_i}$
(1) P は $r \leftarrow \{0,..,p-1\}$ をランダムに選び, $R = g^r$ を計算する. R を V に送信する.
(2) V は $c_i \leftarrow \{0,..,p-1\}(i = 1, 2, \cdots, d)$ をランダムに選び, P に送信する.
(3) P は $z = w - \sum_i x_i c_i \mod p$ を計算し, V に送信する.
(4) V は $g^z = R\prod_i y_i^{-c_i}$ であるか否かを検証し, 成功した場合に証明を受理する.

プロトコル 2 に示された方式は, Honest Verifier ゼロ知識証明であり, 証明者の知識 $(x_1, x_2, \cdots, x_d)$ を証明している. また, [5] で提案されているとおり, V が c_i たちを送信する代わりに $c \leftarrow \{0,..,p-1\}$ を一つだけ選んで送信し, $c_i = c^i \mod p$ としてもよい. このような方式ではプロトコルの全体が一度に実行されているので, 区別してバッチ実行とよぼう. この方式では証明者の計算量においてべき演算の回数が d によらない定数に軽減されており, また, 通信量については c_i たちを $c_i = c^i \mod p$ などとして単一の c から生成することにすれば, d によらない定数に軽減される.

3. 加法的 NP 関係のバッチ証明

NP 関係 R が与えられたとき, 証明者 P と検証者 V の間のプロトコル π で, 知識健全性をみたす対話証明であるとしよう. つまり, $(x, w) \in R$ について x を π の共通入力とすると, 任意の P^* が V と対話して確率 p で受理されるとき, 任意の $y \in \{0,1\}^*$ について

定義 1. *NP* 関係 R を考える. $W(y) = \{x \in \{0,1\}^* | (y, x) \in R\}$, $W = \cup_y W(y)$ とする. このとき, W が多項式時間の演算アルゴリズムが与えられる可換群で, さらに $x_1 \in W(y_1), x_2 \in W(y_2)$ なる y_1, y_2 のみを入力として $x_1 + x_2 \in W(y)$ となる y を出力する多項式時間アルゴリズムが与えられているとき, R を加法的とよぶ.

R を加法的 NP 関係とし, とくに $W = \mathbb{Z}/p\mathbb{Z}$ となる状況を考えよう. ただし p は十分大きい素数である. $(y, x) \in R$ とする. $\pi(y; x)$ を P と V の間の対話証明として, y を共通入力, x を P の補助入力としよう.

$(y_1, x_1), (y_2, x_2), \cdots, (y_d, x_d) \in R$ について π を用いて次のプロトコル $\pi^d(y_1, y_2, \cdots, y_d; x_1, x_2, \cdots, x_d)$ 提案する.

プロトコル 3.
P の秘密入力: $x_i \in \mathbb{Z}/p\mathbb{Z}(i = 1, 2, \cdots, d)$,
共通入力: $y_i (i = 1, 2, \cdots, d)$
(1) V は $e \leftarrow \{0,..,p-1\}$ をランダムに選び, P に送信する.
(2) プロトコル $\pi(\prod_i y_i^{e^{i-1} \mod p}; \sum_i e^{i-1} x_i \mod p)$ を実行する. π において V が証明を受理する場合, またその場合に限り, V は証明を受理する.

次の定理が成り立つ.

定理 1. π が *(*完全, 統計的, 計算量的*)* ゼロ知識証明ならば, π^d は *(*完全, 統計的, 計算量的*)* ゼロ知識証明である.

証明は省略する.

定理 2. π が知識健全ならば, π^d は知識健全である.

Proof. P^* をプロトコル π^d の P にしたがって通信する確率的チューリング機械としよう. P^* と V の対話を (P^*, V) と書いて, P^* が V に「合格」と出力させる確率を $p(y_1, y_2, .., y_d) =$

$\mathrm{Prob}[(P^*, V) \in \mathrm{Acc}]$ としよう. プロトコル π^d の検証者で, 第一メッセージを e として, あとはプロトコル π の検証者と同一の計算を行うもののことを V_e と書こう.

各 $e \in \{0,1,\cdots,p-1\}$ について V_e を, 次で定義される確率的機械とする: プロトコル π^d における V と同じ計算をするが, ステップ 1 で e を出力する. $\mathcal{E} = \{e|\ \Pr[(P^*, V_e) \in \mathrm{Acc}] > \frac{p(y_1,y_2,\cdots,y_d)}{2}\}$ とおく. $\Pr[e \leftarrow_R \{0,1,\cdots,p-1\} : e \in \mathcal{E}|\ (P^*, V_e) \in \mathrm{Acc}] > \frac{1}{2}$ であることがわかる. プロトコル π の知識抽出機を K_P, と書こう. その動作時間の期待値を $\frac{k^c}{p(y_1,y_2,\cdots,y_d)-\epsilon(k)}$ としよう. ただし ϵ は無視できる関数で, プロトコル π の知識誤りである.

次のとおり P^* オラクル機械を考える.

アルゴリズム 1. (1) $i \leftarrow 1$,

(2) $e \leftarrow \{0,..,p-1\}$ をランダムに選ぶ. P^* と V_e の対話をプロトコル π にしたがって実行し, 受理されなければステップ 2に戻る.

(3) K_{P^*} を実行し, 実行時間が $\frac{4k^c}{p(y_1,y_2,\cdots,y_d)-\epsilon(k)}$ に達したら停止する. もし K_{P^*} がある x で $g^x = \prod_j y_j^{e^{j-1} \bmod p}$ を満たすものを出力していれば次に進む. そうでなければ, ステップ 2に戻る.

(4) $e_i \leftarrow e$, $x_i \leftarrow x$ とおき, $i \leftarrow i+1$ とする. もし $i \leqq d$ ならば, ステップ 2に戻る.

(5) E を $d \times d$ 行列を $E_{i,j} = e_j^{i-1}$ で定義する. $(\hat{x}_1, \hat{x}_2, \cdots, \hat{x}_d) = (x_1, x_2, \cdots, x_d)E^{-1}$ を出力する. ただし E^{-1} は F_p に係数を持つ行列としての逆行列である. もし E^{-1} が求まらなければ, ステップ 1に戻る.

$e \in \mathcal{E}$ であれば, K_P が $\frac{4k^c}{p(y_1,y_2,\cdots,y_d)-\epsilon(k)}$ 時間以内に $g^x = \prod_j y_j^{e^{j-1} \bmod p}$ をみたす x を出力する確率は少なくとも $\frac{1}{2}$ である. 行列 E は $e_1, e_2, \cdots, e_d$ がすべて相異なる場合, またその場合に限って可逆であることに注意する. したがって, 上記のアルゴリズムの実行時間の期待値は, ある定数 c' が存在して, 十分大きい k に対して高々 $\frac{k^{c'}}{p(y_1,y_2,\cdots,y_d)-\epsilon(k)}$ である. ■

4. 応 用 例

$g_1, g_2, \cdots, g_d \in G$ について次の NP 関係 R を考える. $R = \{((y,(x_1,x_2,\cdots,x_d))|\ y = g_1^{x_1} g_2^{x_2} \cdots g_d^{x_d}\}$. この NP 関係 R は加法的である. 次のプロトコルは R についての Honest Verifier ゼロ知識証明を与える.

プロトコル 4.
P の秘密入力: $x_i \in \mathbb{Z}/p\mathbb{Z} (i = 1,2,\cdots,d)$,
共通入力: $y = \prod_i g_i^{x_i}$

(1) P は $r_1, r_2, \cdots, r_d \leftarrow \{0,..,p-1\}$ をランダムに選び, $R = \prod_i g_i^{r_i}$ を計算する. R を V に送信する.

(2) V は $c \leftarrow \{0,..,p-1\}$ をランダムに選び, P に送信する.

(3) P は $i = 1,2,\cdots,d$ について $z_i = r_i - cx_i \bmod p$ を計算し, $(z_1, z_2, \cdots, z_d)$ を V に送信する.

(4) V は $\prod_i g_i^{z_i} = Ry^{-c}$ であるか否かを検証し, 成功した場合に証明を受理する.

プロトコル 3 において π を上記のプロトコルとすると, 次のプロトコルを得る.

プロトコル 5.
P の秘密入力: $x(j)_i \in \mathbb{Z}/p\mathbb{Z} (i,j = 1,2,\cdots,d)$,
共通入力: $y(j) = \prod_i g_i^{x(j)_i}$

(1) V は $e \leftarrow \{0,..,p-1\}$ をランダムに選び, V に送信する.

(2) P は $r_1, r_2, \cdots, r_d \leftarrow \{0,..,p-1\}$ をランダムに選び, $R = \prod_i g_i^{r_i}$ を計算する. R を V に送信する.

(3) V は $c \leftarrow \{0,..,p-1\}$ をランダムに選び, P に送信する.

(4) P は $i = 1,2,\cdots,d$ について $z_i = r_i - c\sum_j e^{j-1}x(j)_i \bmod p$ を計算し, $(z_1, z_2, \cdots, z_d)$ を V に送信する.

(5) V は $\prod_i g_i^{z_i} = R(\prod_j y(j)^{e^{j-1} \bmod p})^{-c}$ であるか否かを検証し, 成功した場合に証明を受理する.

さて, プロトコル 2 においては, 証明すべきインスタンス y_i がすべて共通の g によって $y_i = g^{x_i}$ で定義されていた. これはたとえば Pedersen コミットメント [8] などに関して知識を証明したい場合, このままでは必ずしも十分とはいえない. ところで, プロトコル 5 において $i \neq j$ に対して $x(j)_i = 0$ とおけば次のプロトコルを得る. これは底 g が i に関して必ずしも共通とは限らない状況で知識を証明するシグマプロトコルをバッチ実行する.

プロトコル 6.
P **の秘密入力:** $x_i \in \mathbb{Z}/p\mathbb{Z}(i,j=1,2,\cdots,d)$,
共通入力: $y_i = g_i^{x_i}$

（1） V は $e \leftarrow \{0,..,p-1\}$ をランダムに選び, V に送信する.

（2） P は $r_1, r_2, \cdots, r_d \leftarrow \{0,..,p-1\}$ をランダムに選び, $R = \prod_i g_i^{r_i}$ を計算する. R を V に送信する.

（3） V は $c \leftarrow \{0,..,p-1\}$ をランダムに選び, P に送信する.

（4） P は $i = 1,2,\cdots,d$ について $z_i = r_i - ce^{i-1}x_i \mod p$ を計算し, $(z_1, z_2, \cdots, z_d)$ を V に送信する.

（5） V は $\prod_i g_i^{z_i} = R(\prod_i y_i^{e^{i-1} \mod p})^{-c}$ であるか否かを検証し, 成功した場合に証明を受理する.

5. 部分証明のバッチ証明

Cramer たちは [3] において, インスタンスの一部のみに対する知識を証明するプロトコルを与えた. 二つのインスタンス y_0, y_1 に対して $b \in \{0,1\}$ と $(y_b, x) \in R$ となる x の知識を証明するプロトコルを一例として以下に示す.

プロトコル 7.
P **の秘密入力:** $b \in \{0,1\}, x \in \mathbb{Z}/p\mathbb{Z}$,
共通入力: y_0, y_1 ただし $y_b = g^x$.

（1） P は $r, v, c_{1-b} \leftarrow \{0,..,p-1\}$ をランダムに選び, $R_b = g^r$, $R_{1-b} = g^v y_{1-b}^{c_{1-b}}$ を計算する. (R_0, R_1) を V に送信する.

（2） V は $c \leftarrow \{0,..,p-1\}$ をランダムに選び, P に送信する.

（3） P は $c_b = c - c_{1-b}$ として $z_b = r - c_b x \mod p$ を計算, $z_{1-b} = v$ として, V に (z_0, z_1, c_0, c_1) を送信する.

（4） V は $c_0 + c_1 = c$, $g^{z_0} = R_0 y_0^{-c_0}$, $g^{z_1} = R_1 y_1^{-c_1}$ を検証し, すべて成功した場合に証明を受理する.

$(y(1)_0, y(1)_1)$, $(y(2)_0, y(2)_1)$,.., $(y(d)_0, y(d)_1)$ に対してプロトコル 7 をバッチ実行することを考える. 前節とは異なり, NP 関係 $R^2 = \{((y_0,y_1),(b,x)) | \ y_b = g^x\}$ は加法的構造をもたないから, その構成は明らかではない. ここでは, ある種の条件の下でこのような OR 証明のバッチ実行を考える.

5.1 両立困難なペア

定義 2. R を NP 関係とする. ペア $(y_0, y_1) \in \{0,1\}^* \times \{0,1\}^*$ が R-両立困難であるとは, 任意の確率的チューリング機械について, (y_0, y_1), および $(y_b, x) \in R$ となる x と $b \in \{0,1\}$ を入力として $(y_0, x_0) \in R$ かつ $(y_1, x_1) \in R$ となる y_0, y_1, x_0, x_1 を出力する確率が無視できることである.

定義 3. R を加法的 NP 関係とする. 集合 L が R-両立困難で加法的とは, 言語 L_R を部分群に持つ可換群 G が存在して, L が部分群 $L \subset G \times G$ であり, ランダムに選んだ $y \in L$ が無視できる確率を除いて R-両立困難であることである.

例 1. G を離散対数問題が困難な巡回群, $g \in G$ をその生成元とする. $R = \{(y,x) | \ y = g^x\}$ としよう. ハッシュ関数 $\mathcal{H} : \{0,1\}^* \to G$ をさだめて, $L = \{(y_0, y_1) \in G \times G | \ \exists i \in \{0,1\}^* \ y_0 y_1 = \mathcal{H}(i)\}$ とする. L の元はランダムオラクルモデルのもとで R-両立困難である.

例 2. G を離散対数問題が困難な巡回群, $g \in G$ をその生成元とする. $R = \{(y,x) | \ y = g^x\}$ としよう. $\tau \in G$ に対して関数 $\mathcal{H} : \mathbb{Z} \to G$ を $\mathcal{H}(x) = \tau^x$ でさだめて, $L = \{(y_0, y_1) \in G \times G | \ \exists i \in \mathbb{Z} \ y_0 y_1 = \mathcal{H}(i)\}$ とする. $\tau = g^t$ となる t を求めることが不可能であれば, L は R-両立困難で加法的である.

5.2 プロトコルの構成と証明

$L \subset G \times G$ が R 両立困難な加法的集合で, 確率的アルゴリズム I によって $(y_0, y_1) \in L$ が選ばれるとする. $(y(i)_0, y(i)_1)$ が $i = 1,2,\cdots,d$ についてことごとく I によって L から選ばれているとしよう. すると次のプロトコルは知識のゼロ知識証明を与える. 以下では簡単のため, $R = \{(y,x) | y = g^x\}$ とするが, シグマプロトコルによるゼロ知識証明を持つ一般の加法的 NP 関係についても同様である.

プロトコル 8 (部分証明のバッチ実行).
P の秘密入力: $b(1), b(2), \cdots, b(d) \in \{0,1\}$,
$x(1), x(2), \cdots, x(d) \in \mathbb{Z}/p\mathbb{Z}$,
共通入力: $(y(1)_0, y(1)_1), (y(2)_0, y(2)_1), \cdots (y(d)_0, y(d)_1)$
ただし $y(i)_{b(i)} = g^{x(i)}, i = 1, 2 \cdots, d$.

(1) P は $r, v, c(1)_{1-b(1)}, c(2)_{1-b(2)}, \cdots, c(d)_{1-b(d)} \leftarrow \{0, .., p-1\}$ をランダムに選び,

$$R(1)_{b(1)} = g^r,$$
$$R(1)_{1-b(1)} = g^v y(1)_{1-b(1)}^{c_1 c(1)_{1-b(1)}}$$

$i = 2, 3, \cdots, d$ について

$$R(i)_{b(i)} = R(i-1)_{b(i)},$$
$$R(i)_{1-b(i)} = R(i-1)_{1-b(i)} y(i)_{1-b(i)}^{c_1 - b(i)}$$

を計算する. $(R(d)_0, R(d)_1)$ を V に送信する.

(2) V は $c(1) \leftarrow \{0, .., p-1\}$ をランダムに選び, P に送信する.

(3) P は

$$c(1)_{b(1)} = c(1) - c(1)_{1-b(1)}$$
$$z(1)_{b(1)} = r - c(1)_{b(1)} x(1) \mod p,$$
$$z(1)_{1-b(1)} = v,$$

を計算し $c(1)_0$ を V に送信する.

(4) 以下 $i = 2, 3, .., d$ について繰り返す.

(a) V は $c(i) \leftarrow \{0, .., p-1\}$ をランダムに選び, P に送信する.

(b) P は

$$c(i)_{b(i)} = c(i) - c(i)_{1-b(i)}$$
$$z(i)_{b(i)} = z(i-1)_{b(i)} - c(i)_{b(i)} x(i) \mod p,$$
$$z(i)_{1-b(i)} = z(i-1)_{1-b(i)}$$

を計算し $c(i)_0$ を V に送信する.

(5) P は V に $(z(d)_0, z(d)_1)$ を送信する.

(6) V は $i = 1, 2, \cdots, d$ について $c(i)_1 = c(i) - c(i)_0$ として,

$$g^{z(d)_0} = R(d)_0 \prod_{i=1}^{d} y(i)_0^{-c(i)_0},$$
$$g^{z(d)_1} = R(d)_1 \prod_{i=1}^{d} y(i)_1^{-c(i)_1},$$

を検証する. すべて成功した場合に証明を受理する.

定理 3. L が R 両立困難な加法的集合で, 確率的アルゴリズム I によって $(y(i)_0, y(i)_1) \in L$ が $i = 1, 2, .., d$ についてそれぞれ選ばれるとする. すると, プロトコル 8 は *Honest Verifier* ゼロ知識証明かつ知識健全である.

Proof. $i = 1, 2, \cdots, d$ について $(y(i)_0, y(i)_1) \in L_{R^2}$ と仮定する.

次のとおり P^* オラクル機械を考えると, これはプロトコル 8 における知識抽出機となる.

アルゴリズム 2. (1) $m = d$ とおく.

(2) P^* のランダムテープ T を選ぶ.

(3) P^* と V の間でプロトコル 8 を実行する. V が受理しなければ, ステップ 2 に戻る. V が受理すれば, $i = 1, 2, \cdots, d$ について V が出力した $c(i)$ について $c'(i) = c(i)$ とおく. P^* が出力した $c(i)_0$ を $c'(i)_0$ とおく.

(4) $t = 0$ とおく.

(5) P^* と V の間でプロトコル 8 をふたたび実行する. ただし, V は $i = 1, 2, \cdots, m-1$ については $c(i)$ として $c'(i)$ を, $i = m, m+1, \cdots, d$ については $c(i)$ をランダムに選んで出力する. V が受理しないばあい, $t < \frac{2}{p(y)}$ ならば $t = t+1$ としてステップ 5 に戻る. $t \geqq \frac{2}{p(y)}$ ならばステップ 2 に戻る.

(6) $i = 1, 2, \cdots, d$ について $\Delta c(i) = c(i) - c'(i)$, $\Delta c(i)_0 = c(i)_0 - c'(i)_0$, $\Delta c(i)_1 = (c(i) - c(i)_0) - (c'(i) - c'(i)_1)$ とする.

(7) $b \in \{0,1\}$ を $\Delta c(m)_b \neq 0$ となるようえらび,

$$\hat{x}(m) = \frac{1}{\Delta c(m)_b} \left(\Delta z(d)_b - \sum_{j=m+1}^{d} \Delta c(j)_b \hat{x}(j) \right)$$
$$\hat{b}(m) = b$$

とおく.

(8) $y(m)_{\hat{b}(m)} = g^{\hat{x}(m)}$ でなければ中断する.

(9) $m = m-1$ とする. もしも $m > 0$ ならステップ 2 に戻る.

(10) $(\hat{b}(1), \hat{x}(1)), (\hat{b}(2), \hat{x}(2)), \cdots (\hat{b}(d), \hat{x}(d))$ を出力する.

$\Delta c(m)_0 + \Delta c(m)_1 = \Delta c(m)$ が成り立ち, 圧倒的な確率で $\Delta c(m) \neq 0$ が成り立つので, ステップ 7 のような b を少なくともひとつ選ぶことができる.

ステップ 8 で中断する確率が無視できることを証明するために, まず次の事実を証明しよう : $j = m+1, m+2, \cdots d$ について $\Delta c(j)_{1-\hat{b}(j)} = 0$. まず, $j = m+1, m+2, \cdots, d$ について $g^{x(j)} = g^{\hat{x}(j)}, b(j) = \hat{b}(j)$ と仮定して一般性を失わない. さて, $u \in \{m+1, m+2, \cdots, d\}$ が存在して $c = \Delta c(u)_{1-\hat{b}(u)} \neq 0$ がとれるとしよう.

$$A = \sum_{j=m+1, b=\hat{b}(j)}^{d} \Delta c(j)_b \hat{x}(j)$$

とおく. $b \neq b(m)$ の場合,

$$x_b = c^{-1}(\Delta z(d)_b - A),$$
$$x_{1-b} = c^{-1}(\Delta c(j)_b x(m) + \sum_{j=m+1, b \neq \hat{b}(j)}^{d} \Delta c(j)_b \hat{x}(j))$$

とおくと,

$$y_0 = \prod_{j=m, b \neq \hat{b}(j)}^{d} y(d)_0^{\Delta c(j)_b c^{-1}},$$

$$y_1 = \prod_{j=m, b \neq \hat{b}(j)}^{d} y(d)_1^{\Delta c(j)_b c^{-1}},$$

に対して $y_0 = g^{x_0}, y_1 = g^{x_1}$ を満たす．したがって L の R 両立困難性と加法性に矛盾する．$b = b(m)$ の場合，

$$x_b = c^{-1}(\Delta z(d)_b - A - \Delta c(j)_b x(m)),$$

$$x_{1-b} = c^{-1} \sum_{j=m+1, b \neq \hat{b}(j)}^{d} \Delta c(j)_b \hat{x}(j)$$

とおくと，$y_0 = g^{x_0}, y_1 = g^{x_1}$ を満たす．したがって L の R 両立困難性と加法性に矛盾する．

したがって，$b \in \{0,1\}$ についてもし $\Delta c(m)_b \neq 0$ であれば，

$$y(m)_b = g^{\frac{1}{\Delta c(m)_b}\left(\Delta z(d)_b - \sum_{j=m+1}^{d} \Delta c(j)_b \hat{x}(j)\right)}$$

である．とくに，$(y(m)_0, y(m)_1) \in L$ だから $\Delta c(m)_{1-b} = 0$ である．よって $g^{x(m)} = g^{\hat{x}(m)}, b(m) = \hat{b}(m)$ が成り立つ．

■

6. 結論と課題

加法的 NP 関係についてある種の対話証明をバッチ実行するプロトコルの構成方法を示し，安全性を証明した．提案したバッチ実行プロトコルにおいて検証者のチャレンジをハッシュ関数で置き換えることはたやすい．しかしながら，そうして構成されたプロトコルでは，バッチ実行による効率の向上は劇的な場合もあれば，定数倍にとどまる場合もあることがわかった．

効率の向上が限定的である場合についての改良は今後の課題である．劇的な改善が得られなかった理由，またそのような事情が本質的であるのかについては，興味深い問題である．

文　献

[1] M. Bellare, J. Garay, and T. Rabin, "Fast Batch Vefirication for Modular Exponentiation and Digital Signatures," Advances in Cryptology — EUROCRYPTO '98, LNCS 1403, pp. 236–250, Springer-Verlag, 1998.

[2] R. Cramer, "Modular Design of Secure yet Practical Cryptographic Protocols," Ph.D.-thesis, CWI and Uni.of Amsterdam, 1996.

[3] R. Cramer, I. Damgård, and B. Schoenmakers, "Proofs of partial knowledge and simplified design of witness hiding protocols," CRYPTO '94, LNCS 839, pp. 174–187, Springer-Verlag, 1994.

[4] A. Fiat, "Batch RSA," Journal of Cryptology, Vol. 10, pages 75–88, 1997.

[5] R. Gennaro, D. Leigh, R. Sundaram, and W. Yerazunis, "Batching Schnorr identification scheme with applications to privacy-preserving authorization and low-bandwidth communication devices," ASIACRYPT 2004, LNCS 3329, pp. 276–292, Springer-Verlag, 2004.

[6] O. Goldreich, "Foundations of Cryptography," volume I, Cambridge University Press, 2001.

[7] D. M'Raihi, and D. Naccache, "Batch exponentiation: a fast DLP-based signature generation strategy," Proceedings of the 3rd ACM conference on Computer and communications security, 1996

[8] T.P. Pedersen, "Non-interactive and information-theoretic secure verifiable secret sharing," Advances in Cryptology — CRYPTO '91, LNCS 576, pp. 129–140, Springer-Verlag, 1992.

[9] C. Schnorr, "Efficient signature generation by smart cards," Journal of Cryptology, Vol. 4, pages 161–174, 1991.