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

2006－CSEC－34 (18)
2006／7／20

バッチ処理による秘匿回路計算の高速化

千田 浩司† 山本 剛†

† NTT 情報流通プラットフォーム研究所 〒239-0847 神奈川県横須賀市光の丘 1-1
E-mail: †{chida.koji,yamamoto.go}@lab.ntt.co.jp

あらまし マルチパーティプロトコルで実行可能な秘匿回路計算技術は，計算・通信コストの削減が実用化に向けた大きな課題である．本課題に対して，従来では論理回路の基本演算に対するコスト削減が主な研究対象であったが，本研究では，複数の基本演算をバッチ処理してコスト削減する手法について考察する．考察の結果，既存のバッチ処理技術の組み合わせによって，複数の基本演算を個別に処理するよりも計算・通信コストをそれぞれ最大で 38% 及び 27% 程度削減出来る事が分かった．

キーワード 秘匿回路計算，バッチ処理，マルチパーティプロトコル

Efficient Implementation of Secure Circuit Evaluation Using Batch Processing

Koji CHIDA† and Go YAMAMOTO†

† NTT Information Sharing Platform Laboratories Hikarino-oka 1-1, Yokosuka, Kanagawa, 239-0847 Japan
E-mail: †{chida.koji,yamamoto.go}@lab.ntt.co.jp

Abstract Reducing computation and communication costs for multiparty *secure circuit evaluation* (SCE) protocols is a critical issue toward the practical use of the protocols. In this paper, we present some valuable batch processing protocols for existing SCE protocols. The proposed protocols reduce computation and communication costs for existing SCE protocols by up to 38% and 31%, respectively.

Key words Secure Circuit Evaluation, Batch Processing, Multiparty Protocol

1. はじめに

"秘匿回路計算 (Secure Circuit Evaluation)" とは，論理回路における入力値や，回路を構成する基本演算 (論理ゲート) の演算子を秘匿したまま処理実行を可能とする暗号応用技術である．秘匿回路計算を実行するためには，先ず実行対象となる関数を論理回路で表現し，次にその論理回路への入力値，または論理回路を構成する各演算子をエンコードする．但し入力値や演算子はエンコードされた状態のまま実行可能であり，エンコードされた状態から入力値や演算子が有意な確率で推測されないようにする必要がある．なお本技術は一般にソフトウェアで実現され，耐タンパハードウェアなど特殊なハードウェア条件は仮定しない．

秘匿回路計算技術の主な研究動向は 3. 節で詳しく述べるが，汎用的に利用可能であり，かつ入力値や演算子といった秘密情報の秘匿性を保証出来る方式は，我々の知る限りではマルチパーティプロトコルを用いた手法以外は報告されていない．マルチパーティプロトコルでは複数の主体の存在を前提とし，その中から閾値数以上の主体が協力してはじめて秘密情報がエンコードされた論理回路を実行出来るようになる．閾値数以上の主体が協力すると秘密情報を復元出来るが，閾値数未満の主体の結託には耐性があり，秘密情報の秘匿性を何らかの現実的な仮定に基づき保証した方式が幾つか知られている．そのため，アプリケーションによっては主体の総数や閾値数を十分大きくした方が望ましい場合が有り得るだろう．

既存の秘匿回路計算技術には実行対象となる論理回路やその入力に関する制限は特に無い方式が幾つか存在するが，一般にそのような方式は実用的な論理回路を考えると計算・通信コストが膨大となる問題がある．例えば，[28] で提案されたマルチパーティプロトコルを用いた秘匿回路計算（以降，マルチパーティプロトコルを用いた秘匿回路計算をマルチパーティ秘匿回路計算と呼ぶ）を実装した我々のグループの評価結果 [9] によれば，主体の総数及び閾値数を 2 としたとき，各主体の計算機をインテル®Xeon®プロセッサー, CPU クロック 2.8 GHz 4 台として，1 ビット 2 入力の排他的論理和 (XOR) の計算を約 6.5 ミリ秒で処理可能である(注1)．これは例えば共通鍵暗号ア

(注1)：実装報告は XOR のみだが，理論上は任意の 1 ビット 2 入力の論理演

ルゴリズム DES (Data Encryption Standard) で暗号化されたデータに対して秘密鍵を秘匿したまま復号する事を考えたとき(注2)，前記の環境で暗号文 1 ブロック (64 ビット) の復号に約 50 秒掛かる計算となり実用レベルの性能とは言い難い．そこで本研究では，マルチパーティ秘匿回路計算技術の実用化に向けて，同技術の計算・通信コストを削減する手法について考察する．

2. 本研究の成果

本研究では秘匿回路計算の従来のアプローチとは異なり，複数の基本演算 (論理ゲート) が並列処理可能な状況を仮定して，基本演算に必要なゼロ知識証明をバッチ処理する事を試み，その実現可能性について考察した．ここでバッチ処理とは，ある複数の処理が与えられたとき，それを個別に処理するよりも計算コストまたは通信コストに優れ，かつ全ての処理目的を達成出来るような手段全般を指す．また前記の複数の基本演算を並列処理可能とする仮定は，実用的な論理回路を考えれば十分現実的であるといえよう．

考察の結果，Schoenmakers, Tulys が提案したマルチパーティ秘匿回路計算 [27] に対して，同一基底の複数入力に対して適用可能なバッチ処理 [17]，及び異なる基底の複数入力に対して適用可能なバッチ処理 [29] の組み合わせによって，ゼロ知識証明におけるべき演算の回数及び通信データ量をそれぞれ 48% 及び 60% 削減出来る事が分かった．これは [27] のマルチパーティ秘匿回路計算の計算・通信コストについてそれぞれ 38% 及び 23% 程度の削減効果が見込める計算となる．また，我々のグループが [28] において提案したマルチパーティ秘匿回路計算に，部分証明 [10] に対して適用可能なバッチ処理 [29] を適用する事で，ゼロ知識証明におけるべき演算の回数及び通信データ量をそれぞれ 40% 及び 75% 削減出来る事が分かった．これは [28] のマルチパーティ秘匿回路計算の計算・通信コストについてそれぞれ 32% 及び 27% 程度の削減効果が見込める計算となる．

3. 研究動向

秘匿回路計算技術は 1982 年に Yao が提案した方式が起源であると考えられる [30](注3)．Yao は，ハッシュ関数を用いて各論理ゲートをエンコードする手法を提案し，回路 C をエンコードする主体 P_1 と C をエンコードした回路 C' を実行する主体 P_2 の 2 者モデル，即ち 2 パーティプロトコルを確立した．[30] の方式では，P_2 が C' を実行する際，P_2 は P_1 からエンコードされた入力値を得る必要があるが，P_2 は入力値を P_1 に知られないようにするため，P_1 と対話的に紛失通信プロトコル (例えば [12]) を実行し，エンコードされた入力値を得る必要がある．[30] の方式は，エンコードされた論理ゲートの数に比例したハッシュ計算を行う事で実行可能であり，その構成は比較的単純である．2004 年には実装報告がなされている [19]．

[30] の方式では回路や処理結果の正当性は保証されないが，その後の研究で，各主体の不正を検出可能とする方式や [19], [20]，非対話で 2 パーティ秘匿回路計算を実行可能とする手法が提案されている [3], [25]．しかしこれらは一般に計算・通信コストとのトレードオフの関係にある．

一方，エンコードされた入力値または演算子に対して 3 者以上の協力により実行可能なマルチパーティプロトコルも数多く提案されている．例えば [18] では，入力値のエンコードとしてランダムな XOR 分割を用い，紛失通信プロトコル及び検証可秘密分散 (例えば [13], [23]) と組み合わせてマルチパーティプロトコルを実現している ([21], pp. 233-236)．また [16] では，入力値のエンコードとして秘密分散を用いる事でマルチパーティプロトコルを実現している．[16] の方式は passive な攻撃者に対してのみ安全性が保証されるが，阿部は active な攻撃者に対しても安全性を保証出来るようにプロトコルを改良した [1]．なお [1], [18] の方式は，各主体の計算コスト (べき演算の回数) 及び通信コスト (通信データ量) は主体数の増加に合わせて増大し，また各主体が予め入力値と同程度のサイズの秘密情報を保持する必要があるため，入力データ長が大きいほど管理コストが増大するという課題がある．

Franklin, Haber は，閾値準同型暗号を用いたマルチパーティプロトコルを提案した [15]．これを用いてマルチパーティ秘匿回路計算を実行する場合，各主体の計算・通信コストは主体数に関わらず一定であり，保持する秘密情報も一定長となる．[15] の方式は passive な攻撃者に対してのみ安全性が保証されるが，その後 Cramer らは active な攻撃者に対しても安全性を保証出来る，閾値準同型暗号を用いたマルチパーティプロトコルを提案した [11]．[11] の方式は [15] のそれと同様，加法準同型暗号の利用を前提とし，Paillier 暗号 [22] や Franklin-Haber 暗号 [15] に基づく具体例を挙げている．

現在まで知られている加法準同型暗号は素因数分解問題の困難性を安全性の根拠としているが，Schoenmakers, Tulys は離散対数問題に基づく ElGamal 暗号を変形して加法準同型性を持たせ，それを用いたマルチパーティプロトコルを実現した [27]．これにより鍵生成の簡略化(注4)や，楕円曲線を用いた処理高速化が期待出来る．但し，[11], [15] の方式は理論上は全体の処理時間が主体数に関わらず一定と出来るが，[27] の方式は主体数に比例して全体の処理時間が増大するという課題がある．また [27] は optimistic プロトコルであるため，何れかの主体の異常／不正処理があった場合に特別な手続きが必要となり計算・通信コストが掛かる．またそのための設計コストも無視出来ない．

算についても同様の処理時間と見なせる．なお否定 (NOT) の計算はマルチパーティプロトコルを実行する必要が無く十分高速である．またハイプライン処理を効果的に実装する事で，主体の総数及び閾値数に関わらず処理時間をほぼ一定と出来る事を実測値より確認している．

(注2)：具体的なシナリオについては [9] を参照されたい．

(注3)：秘匿回路計算の関連研究として，1978 年に Rivest らがデータを暗号化したまま四則演算可能な暗号方式を発表しているが [24]，その後 Brickell らによって具体的な攻撃方法が報告されている [6]．

(注4)：現在まで，離散対数問題または素因数分解問題に基づく暗号系に対する様々なマルチパーティ鍵生成プロトコルが提案されているが，素因数分解問題に基づくマルチパーティの方が明らかに処理が複雑であり，計算・通信コストが掛かる．

我々のグループは[28]において,[27]の方式をnon-optimisticとし,更に計算・通信コストを[27]の方式よりも削減した手法を提案した.[28]の方式は[27]のそれと同様,全体の処理時間が主体数に比例して増大するが,我々はパイプライン処理を用いてその増加率を殆ど無視出来る事を実装により確認した[9].

最後に上述の各種マルチパーティ秘匿回路計算技術について,それぞれの特徴を表1にまとめておく.

4. 関連技術

第2.節でも述べたとおり,本研究ではマルチパーティ秘匿回路計算の計算・通信コスト削減を目的として,複数のゼロ知識証明についてバッチ処理する事を考える.既存のマルチパーティ秘匿回路計算に適用可能なバッチ処理を第4.1, 4.2, 4.3節で概略説明する.なお以降 $\mathcal{G}=\langle g\rangle$ を素数位数 p の可換群,$\mathcal{H}(\cdot)$ を任意のビット列を $\mathbb{Z}/p\mathbb{Z}$ に写す一方向性ハッシュ関数,P, V をそれぞれ証明者,検証者とする.

4.1 同一基底に対するバッチ処理

Gennaroらは[17]において,Schnorr認証プロトコルを用いて単一の証明者(被認証者)が複数の異なるIDに対する知識を証明する際,バッチ処理により効果的に証明する手法を提案した.以下は[17]の方法に従って,$i=1,\ldots,d$ について共通入力 $y_i=g^{x_i}\in\mathcal{G}$ に対する $x_i\in\mathbb{Z}/p\mathbb{Z}$ の知識をFiat-Shamir heuristics[14]を用いて非対話で証明するプロトコルである.

> [プロトコル1](文献[17])
> **P の秘密入力:** $x_i\in\mathbb{Z}/p\mathbb{Z}$
> **共通入力:** $\mathcal{G}, p, g, y_i=g^{x_i}\in\mathcal{G}$ $\quad(i=1,\ldots,d)$
>
> (1) P は $r\in_R\mathbb{Z}/p\mathbb{Z}$ を選び,
> $R=g^r$, $c=\mathcal{H}(R)$, $z=r-\sum_i x_i c^i \bmod p$,
> を計算し,(c,z) を V に送信する.
> (2) V は $c=\mathcal{H}(g^z\prod_i y_i^{c^i})$ が成り立てば P の証明を受理する.

[プロトコル1]は,$d=1$ とすれば良く知られる知識証明プロトコル(Schnorr署名[26]においてメッセージを空の値とした場合に等しい)だが,$d=1$ として[プロトコル1]を単純に α 回実行するよりも,$d=\alpha$ として[プロトコル1]を実行する方が P, V のべき演算の回数,及び P から V へ送信される通信データ量を大幅に削減出来る事は明らかである.

4.2 異なる基底に対するバッチ処理

次に $i=1,\ldots,d$ について共通入力 $(g_i, y_i=g_i^{x_i})\in\mathcal{G}^2$ に対する $x_i\in\mathbb{Z}/p\mathbb{Z}$ の知識を非対話で証明するプロトコルについて考える.今証明者が $i\neq j$ となる全ての $i,j\in[1,\ldots,d]$ について $\log_{g_i} g_j$ を知らなければ,[プロトコル1]を単純に適用する事は困難である.我々は[29]において,上述のプロトコルに適用可能なバッチ処理を提案した.以下でそのプロトコルを紹介する.

> [プロトコル2](文献[29])
> **P の秘密入力:** $x_i\in\mathbb{Z}/p\mathbb{Z}$
> **共通入力:** $\mathcal{G}, p, (g_i, y_i=g_i^{x_i})\in\mathcal{G}^2$ $\quad(i=1,\ldots,d)$
>
> (1) P, V は $e=\mathcal{H}(g_1\|\cdots\|g_d\|y_1\|\cdots\|y_d)$ を計算する.
> (2) P は $i=1,\ldots,d$ について $r_i\in_R\mathbb{Z}/p\mathbb{Z}$ を選び,
> $R=\prod_i g_i^{r_i}$, $c=\mathcal{H}(R)$, $z_i=r_i-e^{i-1}cx_i \bmod p$,
> を計算し,$(c,z_1,\ldots,z_d)$ を V に送信する.
> (3) V は $c=\mathcal{H}(\prod_i g_i^{z_i}y_i^{e^{i-1}c})$ が成り立てば P の証明を受理する.

[プロトコル2]は,$d=e=1$ とすれば良く知られる知識証明プロトコル証明だが,$d=e=1$ として[プロトコル2]を α 回実行するよりも,$d=\alpha$ として[プロトコル2]を実行する方が P から V へ送信される通信データ量を $(\alpha+1)/2\alpha$ 倍に削減出来る.なお,べき演算の回数は P, V ともに削減されないが,個々のべき演算結果を必要とする従来の方法に対して,[プロトコル2]は P, V ともに全てのべき演算の積を求めれば良い事が分かる.すると[4],[7]等で提案されている,べき演算の積を効果的に計算するアルゴリズムを実行する事で処理高速化が見込める.

4.3 部分証明に対するバッチ処理

我々は[29]において,Cramerらが提案した部分証明(Partial Proof)[10]をバッチ処理するプロトコルを提案した.部分証明は,ある二つの異なる共通入力に対して,どちらか片方の共通入力に対する秘密情報の知識を証明するプロトコルである.例えば共通入力 $y_0=g^{x_0}\in\mathcal{G}$ 及び $y_1=g^{x_1}\in\mathcal{G}$ に対する秘密情報 $(b,x_b)\in\{0,1\}\times\mathbb{Z}/p\mathbb{Z}$ の知識を証明出来る.即ち何れか片方の知識さえあれば証明可能となる.また,ある二つの異なる共通入力に対して,少なくともどちらか片方の共通入力がある言語に属している事を証明する手法も知られている(例えば[2]).

[29]に従って $i=1,\ldots,d$ について共通入力 $(y_{i,0}, y_{i,1})=(g^{x_{i,0}}, g^{x_{i,1}})\in\mathcal{G}^2$ に対する $(b_i, x_{i,b_i})\in\{0,1\}\times\mathbb{Z}/p\mathbb{Z}$ の知識を非対話で証明するプロトコルを[プロトコル8]として付録に記載する.以下は[プロトコル8]を言語証明に拡張したプロト

表 1　既存のマルチパーティ秘匿回路計算技術の特徴 (n: 主体数. 処理単位は基本演算.)

	計算コスト	通信コスト	処理時間	エンコード	備考
[18]	$O(n^3)$	$O(n^3)$	$O(1)$	ランダム XOR 分割	
[16]	—	—	$O(1)$	秘密分散	active 攻撃者に対して安全で無い
[1]	$O(n^2)$	$O(n^2)$	$O(1)$	秘密分散	[16] を改良し active 攻撃者に対しても安全
[15]	$O(n)$	$O(n)$	$O(1)$	素因数分解問題に基づく加法準同型暗号	active 攻撃者に対して安全で無い
[11]	$O(n)$	$O(n)$	$O(1)$	素因数分解問題に基づく加法準同型暗号	[15] を改良し active 攻撃者に対しても安全
[27]	$O(n)$	$O(n)$	$O(n)$	楕円 ElGamal 暗号	optimistic
[9],[28]	$O(n)$	$O(n)$	$O(n)$	楕円 ElGamal 暗号	・[27] を改良し non-optimistic かつ 一定倍計算・通信コスト削減 ・パイプライン処理により処理時間の増加率を大幅に削減

コルである.

[プロトコル 3] (文献 [29])

P の秘密入力: $(b_i, x_{i,b_i} = x'_{i,b_i}) \in \{0,1\} \times \mathbb{Z}/p\mathbb{Z}$

共通入力: $\mathcal{G}$, p, g, $h \in \mathcal{G}$,

$$(y_{i,0}, y'_{i,0}, y_{i,1}, y'_{i,1}) = (g^{x_{i,0}}, h^{x'_{i,0}}, g^{x_{i,1}}, h^{x'_{i,1}}) \in \mathcal{G}^2 \quad (i = 1, \ldots, d)$$

(1) P は $r, v, c_{i,1-b_i} \in_R \mathbb{Z}/p\mathbb{Z}$ を選び $(i = 1, \ldots, d)$,

$R_0 = g^r \prod_{\{i|b_i=1\}} y_{i,0}^{c_{i,0}}$, $S_0 = h^r \prod_{\{i|b_i=1\}} y'^{c_{i,0}}_{i,0}$,

$R_1 = g^v \prod_{\{i|b_i=0\}} y_{i,1}^{c_{i,1}}$, $S_1 = h^v \prod_{\{i|b_i=0\}} y'^{c_{i,1}}_{i,1}$,

を計算し, $i = 1, \ldots, d$ について順に

$c_i = \mathcal{H}(c_{i-1} \| c_{i-1,0})$, $c_{i,b_i} = c_i - c_{i,1-b_i} \bmod p$,

を計算し (但し $c_0 = (R_0 \| S_0)$, $c_{0,0} = (R_1 \| S_1)$), その後

$z_0 = r - \sum_{\{i|b_i=0\}} c_{i,0} x_{i,0} \bmod p$,

$z_1 = v - \sum_{\{i|b_i=1\}} c_{i,1} x_{i,1} \bmod p$,

を計算して $(c_1, c_{1,0}, \ldots, c_{d,0}, z_0, z_1)$ を V に送信する.

(2) V は $i = 1, \ldots, d$ について順に

$c_{i,1} = c_i - c_{i,0} \bmod p$, $c_{i+1} = \mathcal{H}(c_i \| c_{i,0})$,

を計算し (但し c_{d+1} は計算不要),

$$c_1 = \mathcal{H}(g^{z_0} \prod_i y_{i,0}^{c_{i,0}} \| h^{z_0} \prod_i y'^{c_{i,0}}_{i,0} \| g^{z_1} \prod_i y_{i,1}^{c_{i,1}} \| h^{z_1} \prod_i y'^{c_{i,1}}_{i,1}),$$

が成り立てば P の証明を受理する.

[プロトコル 3] は, $d = 1$ とすれば [10] の部分証明を言語証明に拡張したプロトコルとなるが, $d = 1$ として [プロトコル 3] を α 回実行するよりも, $d = \alpha$ として [プロトコル 3] を実行する方が P, V のべき演算の回数, 及び P から V へ送信される通信データ量を削減出来る. 具体的には, P, V のべき演算の回数はそれぞれ $(\alpha + 2)/3\alpha$ 倍, $(\alpha + 1)/2\alpha$ 倍に, 通信データ量は $(\alpha + 3)/4\alpha$ 倍に削減出来る.

5. バッチ処理の適用可能性考察

本節では既存のマルチパーティ秘匿回路計算へのバッチ処理の適用可能性について, 計算・通信コストが比較的小さいマルチパーティ秘匿回路計算技術 [27], [28] を対象として考察する. [27], [28] の方式については本稿では説明を省略するが, [27], [28] の方式を用いた基本演算はそれぞれ以下のプロトコルを実行する.

[プロトコル 4] (文献 [27])

P の秘密入力: $(s, t, w, x) \in (\mathbb{Z}/p\mathbb{Z})^4$

共通入力: $\mathcal{G}$, p, g, $(h, h') \in \mathcal{G}^2$, $(G, H, X, Y) \in \mathcal{G}^4$,

$(Z, \bar{G}, \bar{H}, \bar{X}, \bar{Y}) = (g^x h'^w, g^s G^x, h^s H^x, g^t X^x, h^t Y^x) \in \mathcal{G}^5$

(1) P は $\delta, \eta, \rho, \tau \in_R \mathbb{Z}/p\mathbb{Z}$ を選び,

$A = g^\eta h'^\rho$, $B = g^\delta G^\eta$, $C = h^\delta H^\eta$,

$D = g^\tau X^\eta$, $E = h^\tau Y^\eta$,

を計算し, その後

$c = \mathcal{H}(A \| B \| C \| D \| E)$,

$z_1 = \eta - cx \bmod p$, $z_2 = \rho - cw \bmod p$,

$z_3 = \delta - cs \bmod p$, $z_4 = \tau - ct \bmod p$,

を計算して (c, z_1, z_2, z_3, z_4) を V に送信する.

(2) V は以下が成り立てば P の証明を受理する.

$$c = \mathcal{H}(g^{z_1} h'^{z_2} Z^c \| g^{z_3} G^{z_1} \bar{G}^c \| h^{z_3} H^{z_1} \bar{H}^c \| g^{z_4} X^{z_1} \bar{X}^c \| h^{z_4} Y^{z_1} \bar{Y}^c).$$

[プロトコル 5] (文献 [28])

P の秘密入力: $(b, s_b = s'_b, t_b = t'_b) \in \{0,1\} \times (\mathbb{Z}/p\mathbb{Z})^2$

共通入力: $\mathcal{G}$, p, g, $h \in \mathcal{G}$,

$$(G_0, H_0, G_1, H_1, X_0, Y_0, X_1, Y_1) = (g^{s_0}, h^{s'_0}, g^{s_1}, h^{s'_1}, g^{t_0}, h^{t'_0}, g^{t_1}, h^{t'_1}) \in \mathcal{G}^8$$

(1) P, V は以下を計算する.

$c = \mathcal{H}(g \| h \| G_0 \| H_0 \| G_1 \| H_1 \| X_0 \| Y_0 \| X_1 \| Y_1)$.

(2) P は $r, v, c_{1-b} \in_R \mathbb{Z}/p\mathbb{Z}$ を選び,

$R_b = g^r$, $S_b = h^r$, $R_{1-b} = g^v (G_{1-b} X_{1-b}^c)^{c_{1-b}}$,

$S_{1-b} = h^v (H_{1-b} Y_{1-b}^c)^{c_{1-b}}$,

$c = \mathcal{H}(R_0 \| S_0 \| R_1 \| S_1)$, $c_b = c - c_{1-b} \bmod p$,

$z_b = r - c_b(s_b + e t_b) \bmod p$, $z_{1-b} = v \bmod p$,

を計算し, (z_0, z_1, c, c_0) を V に送信する.

(3) V は $c_1 = c - c_0 \bmod p$ を計算し, 以下が成り立てば P の証明を受理する.

$$c = \mathcal{H}(g^{z_0} (G_0 X_0^c)^{c_0} \| h^{z_0} (H_0 Y_0^c)^{c_0} \| g^{z_1} (G_1 X_1^c)^{c_1} \| h^{z_1} (H_1 Y_1^c)^{c_1}).$$

すると [プロトコル 4] について, [プロトコル 1, 2] の手続き

を組み合わせて以下のバッチ処理を構成する事が出来る.

[プロトコル 6]
P の秘密入力: $(s_i, t_i, w_i, x_i) \in (\mathbb{Z}/p\mathbb{Z})^4$
共通入力: $\mathcal{G}$, p, g, $(h, h') \in \mathcal{G}^2$, $(G_i, H_i, X_i, Y_i) \in \mathcal{G}^4$,
$(Z_i, \bar{G}_i, \bar{H}_i, \bar{X}_i, \bar{Y}_i)$
$= (g^{x_i} h'^{w_i}, g^{s_i} G_i^{x_i}, h^{s_i} H_i^{x_i}, g^{t_i} X_i^{x_i}, h^{t_i} Y_i^{x_i}) \in \mathcal{G}^5$
$(i = 1, \ldots, d)$

(1) P, V は
$e = \mathcal{H}(g \| h \| h' \| \mathbf{G} \| \mathbf{H} \| \mathbf{X} \| \mathbf{Y} \| \mathbf{Z} \| \bar{\mathbf{G}} \| \bar{\mathbf{H}} \| \bar{\mathbf{X}} \| \bar{\mathbf{Y}})$
を計算する. 但し任意の bold 体 $\mathbf{A}$ について $\mathbf{A} = (A_1 \| \cdots \| A_d)$ とする.
(2) P は $\delta, \eta_i, \rho_i, \tau \in_R \mathbb{Z}/p\mathbb{Z}$ を選び $(i = 1, \ldots, d)$,
$A = g^{\sum_i \eta_i} h'^{\sum_i \rho_i}$, $B = g^{\delta} \prod_i G_i^{e^{i-1}\eta_i}$,
$C = h^{\delta} \prod_i H_i^{e^{i-1}\eta_i}$, $D = g^{\tau} \prod_i X_i^{e^{i-1}\eta_i}$,
$E = h^{\tau} \prod_i Y_i^{e^{i-1}\eta_i}$,
を計算し, その後
$c = \mathcal{H}(A \| B \| C \| D \| E)$, $z_{1,i} = \eta_i - e^{i-1} c x_i \bmod p$,
$z_{2,i} = \rho_i - e^{i-1} c w_i \bmod p$, $z_3 = \delta - c(\sum_i e^{i-1} s_i) \bmod p$,
$z_4 = \tau - c(\sum_i e^{i-1} t_i) \bmod p$, $(i = 1, \ldots, d)$
を計算して $(c, z_{1,1}, \ldots, z_{1,d}, z_{2,1}, \ldots, z_{2,d}, z_3, z_4)$ を V に送信する.
(3) V は以下が成り立てば P の証明を受理する.
$c = \mathcal{H}(g^{\sum_i z_{1,i}} h'^{\sum_i z_{2,i}} \prod_i Z_i^{e^{i-1}c}$
$\| g^{z_3} \prod_i G_i^{z_{1,i}} \bar{G}_i^{e^{i-1}c} \| h^{z_3} \prod_i H_i^{z_{1,i}} \bar{H}_i^{e^{i-1}c}$
$\| g^{z_4} \prod_i X_i^{z_{1,i}} \bar{X}_i^{e^{i-1}c} \| h^{z_4} \prod_i Y_i^{z_{1,i}} \bar{Y}_i^{e^{i-1}c})$.

[プロトコル 6] は, $d = e = 1$ とすれば [プロトコル 4] と等しいが, [プロトコル 4] を α 回実行するよりも, $d = \alpha$ として [プロトコル 6] を実行する方が P, V のべき演算の回数, 及び P から V へ送信される通信データ量を削減出来る. 具体的な削減効果については第 7. 節で評価する.

一方 [プロトコル 5] については, [プロトコル 3] を適用する事で以下のバッチ処理を構成する事が出来る.

[プロトコル 7]
P の秘密入力:
$(b_i, s_{i,b_i} = s'_{i,b_i}, t_{i,b_i} = t'_{i,b_i}) \in \{0,1\} \times (\mathbb{Z}/p\mathbb{Z})^2$
共通入力: $\mathcal{G}$, p, g, $h \in \mathcal{G}$,
$L_i = (G_{i,0}, H_{i,0}, G_{i,1}, H_{i,1}, X_{i,0}, Y_{i,0}, X_{i,1}, Y_{i,1})$
$= (g^{s_{i,0}}, h^{s'_{i,0}}, g^{s_{i,1}}, h^{s'_{i,1}}, g^{t_{i,0}}, h^{t'_{i,0}}, g^{t_{i,1}}, h^{t'_{i,1}}) \in \mathcal{G}^8$
$(i = 1, \ldots, d)$

(1) P, V は
$e = \mathcal{H}(g \| h \| \mathbf{G_0} \| \mathbf{H_0} \| \mathbf{G_1} \| \mathbf{H_1} \| \mathbf{X_0} \| \mathbf{Y_0} \| \mathbf{X_1} \| \mathbf{Y_1})$ を計算する. 但し任意の bold 体 $\mathbf{A_b}$ について $\mathbf{A_b} = (A_{1,b} \| \cdots \| A_{d,b})$ とする.
(2) P は $r, v, c_{i,1-b_i} \in_R \mathbb{Z}/p\mathbb{Z}$ を選び $(i = 1, \ldots, d)$,
$R_0 = g^r \prod_{\{i|b_i=1\}} (G_{i,0} X_{i,0}^e)^{c_{i,0}}$,
$S_0 = h^r \prod_{\{i|b_i=1\}} (H_{i,0} Y_{i,0}^e)^{c_{i,0}}$,
$R_1 = g^v \prod_{\{i|b_i=0\}} (G_{i,1} X_{i,1}^e)^{c_{i,1}}$,
$S_1 = g^v \prod_{\{i|b_i=0\}} (H_{i,1} Y_{i,1}^e)^{c_{i,1}}$,
を計算し, $i = 1, \ldots, d$ について順に
$c_i = \mathcal{H}(c_{i-1} \| c_{i-1,0})$, $c_{i,b_i} = c_i - c_{i,1-b_i} \bmod p$,
を計算し (但し $c_0 = (R_0 \| S_0)$, $c_{0,0} = (R_1 \| S_1)$), その後
$z_0 = r - \sum_{\{i|b_i=0\}} c_{i,0}(s_{i,0} + e t_{i,0}) \bmod p$,
$z_1 = v - \sum_{\{i|b_i=1\}} c_{i,1}(s_{i,1} + e t_{i,1}) \bmod p$,
を計算して $(c_1, c_{1,0}, \ldots, c_{d,0}, z_0, z_1)$ を V に送信する.
(3) V は $i = 1, \ldots, d$ について順に
$c_{i,1} = c_i - c_{i,0} \bmod p$, $c_{i+1} = \mathcal{H}(c_i \| c_{i,0})$,
を計算し (但し c_{d+1} は計算不要),
$c_1 = \mathcal{H}(g^{z_0} \prod_i (G_{i,0} X_{i,0}^e)^{c_{i,0}} \| h^{z_0} \prod_i (H_{i,0} Y_{i,0}^e)^{c_{i,0}}$
$\| g^{z_1} \prod_i (G_{i,1} X_{i,1}^e)^{c_{i,1}} \| h^{z_1} \prod_i (H_{i,1} Y_{i,1}^e)^{c_{i,1}})$,
が成り立てば P の証明を受理する.

[プロトコル 7] は, $d = e = 1$ とすれば [プロトコル 5] と等しいが, [プロトコル 5] を α 回実行するよりも, $d = \alpha$ として [プロトコル 7] を実行する方が P, V のべき演算の回数, 及び P から V へ送信される通信データ量を削減出来る. 具体的な削減効果については, [プロトコル 6] 同様, 第 7. 節で評価する.

6. セキュリティ考察

6.1 [プロトコル 6] について

[プロトコル 4] を [プロトコル 2] に従って d 組の共通入力に拡張すると以下のプロトコルを考える事が出来る.

[プロトコル A]
P の秘密入力: $(s_i, t_i, w_i, x_i) \in (\mathbb{Z}/p\mathbb{Z})^4$
共通入力: $\mathcal{G}$, p, $(g_i, h_i, h'_i) \in \mathcal{G}^3$, $(G_i, H_i, X_i, Y_i) \in \mathcal{G}^4$,
$(Z_i, \bar{G}_i, \bar{H}_i, \bar{X}_i, \bar{Y}_i)$
$= (g_i^{x_i} h'^{w_i}_i, g_i^{s_i} G_i^{x_i}, h_i^{s_i} H_i^{x_i}, g_i^{t_i} X_i^{x_i}, h_i^{t_i} Y_i^{x_i}) \in \mathcal{G}^5$
$(i = 1, \ldots, d)$

(1) P, V は

$$e = \mathcal{H}(\mathbf{g}\|\mathbf{h}\|\mathbf{h}'\|\mathbf{G}\|\mathbf{H}\|\mathbf{X}\|\mathbf{Y}\|\mathbf{Z}\|\bar{\mathbf{G}}\|\bar{\mathbf{H}}\|\bar{\mathbf{X}}\|\bar{\mathbf{Y}})$$

を計算する．但し任意の bold 体 $\mathbf{A}$ について $\mathbf{A} = (A_1\|\cdots\|A_d)$ とする．

（2） P は $\delta_i, \eta_i, \rho_i, \tau_i \in_R \mathbb{Z}/p\mathbb{Z}$ を選び $(i = 1, \ldots, d)$，

$A = \prod_i g_i^{\eta_i} h_i'^{\rho_i}$, $B = \prod_i g_i^{\delta_i} G_i^{\eta_i}$, $C = \prod_i h_i^{\delta_i} H_i^{\eta_i}$,

$D = \prod_i g_i^{\tau_i} X_i^{\eta_i}$, $E = \prod_i h_i^{\tau_i} Y_i^{\eta_i}$,

を計算し，その後

$c = \mathcal{H}(A\|B\|C\|D\|E)$,

$z_{1,i} = \eta_i - e^{i-1} c x_i \bmod p$, $z_{2,i} = \rho_i - e^{i-1} c w_i \bmod p$,

$z_{3,i} = \delta_i - e^{i-1} c s_i \bmod p$, $z_{4,i} = \tau_i - e^{i-1} c t_i \bmod p$,

を計算して

$$(c, z_{1,1}, \ldots, z_{1,d}, z_{2,1}, \ldots, z_{2,d}, z_{3,1}, \ldots, z_{3,d}, z_{4,1}, \ldots, z_{4,d})$$

を V に送信する．

（3） V は以下が成り立てば P の証明を受理する．

$$c = \mathcal{H}(\textstyle\prod_i g_i^{z_{1,i}} h_i'^{z_{2,i}} Z_i^{e^{i-1}c} \| \prod_i g_i^{z_{3,i}} G_i^{z_{1,i}} \bar{G}_i^{e^{i-1}c} \| \prod_i h_i^{z_{3,i}} H_i^{z_{1,i}} \bar{H}_i^{e^{i-1}c} \| \prod_i g_i^{z_{4,i}} X_i^{z_{1,i}} \bar{X}_i^{e^{i-1}c} \| \prod_i h_i^{z_{4,i}} Y_i^{z_{1,i}} \bar{Y}_i^{e^{i-1}c})$$

すると[29]の定理 1, 2 より以下の補題が成り立つ．

補題 1. [プロトコル 2] 及び [プロトコル 4] がゼロ知識／知識健全であれば，[プロトコル A] はランダムオラクルモデルの下でゼロ知識／知識健全である．

次に [プロトコル A] 及び [プロトコル 6] の関係について考える．いま [プロトコル A] において

$$\begin{cases} g = g_1 = \cdots = g_d, \\ h = h_1 = \cdots = h_d, \\ h' = h_1' = \cdots = h_d', \end{cases} \tag{1}$$

とし，$\delta \leftarrow \sum_i \delta_i$, $\tau \leftarrow \sum_i \tau_i$, $z_3 \leftarrow \sum_i z_{3,i}$, $z_4 \leftarrow \sum_i z_{4,i}$ と置き換えてやれば，これは [プロトコル 6] に他ならない．従って [プロトコル A] において式 (1) を満たせば，これは [プロトコル 6] と等しい事が分かり，以下の命題を得る事が出来る．

命題 1. [プロトコル 2] 及び [プロトコル 4] がゼロ知識／知識健全であれば，[プロトコル 6] はランダムオラクルモデルの下でゼロ知識／知識健全である．

6.2 [プロトコル 7] について

[プロトコル 7] のステップ (1) 実行後，以下の入力を与えたとする．

[入力 2]

P の秘密入力:

$$(b_i, \tilde{s}_{i,b_i} = s_{i,b_i} + e t_{i,b_i} = s'_{i,b_i} + e t'_{i,b_i}) \in (\mathbb{Z}/p\mathbb{Z})^2$$

共通入力: $\mathcal{G}$, p, g, $h \in \mathcal{G}$, $(\tilde{G}_{i,0}, \tilde{H}_{i,0}, \tilde{G}_{i,1}, \tilde{H}_{i,1}) =$

$$(g^{s_{i,0}+e t_{i,0}}, h^{s'_{i,0}+e t'_{i,0}}, g^{s_{i,1}+e t_{i,1}}, h^{s'_{i,1}+e t'_{i,1}}) \in \mathcal{G}^4 \qquad (i = 1, \ldots, d)$$

すると $i = 1, \ldots, d$ について，

$x_{i,0} \leftarrow s_{i,0} + e t_{i,0}$, $x_{i,1} \leftarrow s_{i,1} + e t_{i,1}$,

$y_{i,0} \leftarrow G_{i,0} X_{i,0}^e$, $y'_{i,0} \leftarrow H_{i,0} Y_{i,0}^e$,

$y_{i,1} \leftarrow G_{i,1} X_{i,1}^e$, $y'_{i,1} \leftarrow H_{i,1} Y_{i,1}^e$

と置き換えてやれば，[入力 2] を入力として [プロトコル 7] のステップ (2), (3) を実行するプロトコルは [プロトコル 3] に他ならない．従って，[29] の定理 1, 2 より以下の補題が成り立つ．

補題 2. [プロトコル 3] がゼロ知識／知識健全であれば，[プロトコル 7] はランダムオラクルモデルの下でゼロ知識／知識健全である．

一方[29]から，[プロトコル 3] がゼロ知識かつ知識健全であるためには，[プロトコル 3] の共通入力 $(y_{i,0}, y'_{i,0}, y_{i,1}, y'_{i,1})$ が全ての i について R 両立困難かつ加法的であり，更に各共通入力が独立であれば良い．これは結局，[プロトコル 7] の共通入力 $L_i = (G_{i,0}, H_{i,0}, G_{i,1}, H_{i,1}, X_{i,0}, Y_{i,0}, X_{i,1}, Y_{i,1})$ が全ての i について R 両立困難かつ加法的であり，更に各 L_i が独立であれば，[プロトコル 7] はゼロ知識かつ知識健全である事がいえる．

「R 両立困難」の定義は[29]で述べられておりここでは省略するが，R 両立困難性について以下の補題が成り立つ．

補題 3. $\mathcal{G} = \langle g \rangle$ を素数位数 p の可換群とし，$h \in \mathcal{G}$ とする．このとき，ある $b \in \{0, 1\}$，$s_0, s'_0, s_1, s'_1, t_0, t'_0, t_1, t'_1, \epsilon \in \mathbb{Z}/p\mathbb{Z}$.

$L = (G_0, H_0, G_1, H_1, X_0, Y_0, X_1, Y_1)$

$= (g^{s_0}, h^{s'_0}, g^{s_1}, h^{s'_1}, g^{t_0}, h^{t'_0}, g^{t_1}, h^{t'_1}) \in \mathcal{G}^8$

があって，

$$\begin{cases} \log_g(G_b X_b^\epsilon) = \log_h(H_b Y_b^\epsilon) \\ \log_g(G_{1-b} X_{1-b}^\epsilon) \neq \log_h(H_{1-b} Y_{1-b}^\epsilon) \end{cases}$$

であれば，L は R 両立困難である．

また R 両立困難な入力 L, L' が加法的であるとは，

$L \times L' =$

$(G_0 G'_0, H_0 H'_0, G_1 G'_1, H_1 H'_1, X_0 X'_0, Y_0 Y'_0, X_1 X'_1, Y_1 Y'_1)$

が R 両立困難な事である．

以上の考察から，以下の命題を得る事が出来る．

命題 2. $(k_1, \ldots, k_d) \neq (0, \ldots, 0)$ なる任意の $(k_1, \ldots, k_d) \in \{0, 1\}^d$ について [プロトコル 7] の共通入力 L_i $(i = 1, \ldots, d)$ の積

$$\prod_i L_i^{k_i} = (\prod_i G_{i,0}^{k_i}, \prod_i H_{i,0}^{k_i}, \prod_i G_{i,1}^{k_i}, \prod_i H_{i,1}^{k_i}, \prod_i X_{i,0}^{k_i}, \prod_i Y_{i,0}^{k_i}, \prod_i X_{i,1}^{k_i}, \prod_i Y_{i,1}^{k_i})$$

が

$$\begin{cases} \log_g \prod_i (G_{i,b_i} X_{i,b_i}^\epsilon)^{k_i} = \log_h \prod_i (H_{i,b_i} Y_{i,b_i}^\epsilon)^{k_i} \\ \log_g \prod_i (G_{i,1-b_i} X_{i,1-b_i}^\epsilon)^{k_i} \neq \log_h \prod_i (H_{i,1-b_i} Y_{i,1-b_i}^\epsilon)^{k_i} \end{cases} \tag{2}$$

の関係を満たし，かつ各 L_i が独立であれば，[プロトコル 7] はランダムオラクルモデルの下でゼロ知識かつ知識健全である．

実際[28]のプロトコルの場合について考えると，[プロトコル 7] の共通入力を生成する暗号文の作成者が honest であれば，各 L_i は独立であり，$\prod_i L_i^{k_i}$ は無視出来る確率を除いて式 (2) を満たす．

7. 計算・通信コストの評価

本節では [プロトコル 4, 5, 6, 7] の計算・通信コスト, [27], [28] のプロトコルの計算・通信コスト, 及び [プロトコル 6, 7] をそれぞれ [27], [28] のプロトコルに適用した場合の計算・通信コストについて評価する. 結果を表 2 にまとめる. ここで n, d はそれぞれ主体数, バッチ処理する共通入力の組の数とする. [プロトコル 4, 5] 及び [27], [28] のプロトコルは d 回繰り返すものとする. なお計算コストは, 証明者 P 及び検証者 V のべき演算の回数として評価する. 通信コストは証明者が送信する通信データ量とし, 単位はビットとする.

表 2 各プロトコルの計算・通信コストの比較

	P の計算コスト	V の計算コスト	通信コスト
[プロトコル 4] (**P4**)	$10d$	$15d$	$5d\|p\|$
[プロトコル 5] (**P5**)	$8d$	$12d$	$4d\|p\|$
[プロトコル 6] (**P6**) (**P4** のバッチ処理)	$4d+6$	$9d+6$	$(2d+3)\|p\|$
[プロトコル 7] (**P7**) (**P5** のバッチ処理)	$4d+4$	$8d+4$	$(d+3)\|p\|$
[27]	$18dn$	$19dn$	$(6\|g\|+7\|p\|)dn$
[28]	$15dn$	$16dn$	$(5\|g\|+6\|p\|)dn$
[27] に **P6** 適用 ([27] のバッチ処理)	$(10d+6)n$	$(13d+6)n$	$(6d\|g\|+(4d+3)\|p\|)n$
[28] に **P7** 適用 ([28] のバッチ処理)	$(9d+4)n$	$(12d+4)n$	$(5d\|g\|+(3d+3)\|p\|)n$

表 2 から, d が十分大きければ, [プロトコル 6] は [プロトコル 4] と比べて証明者はおよそ 60%, 検証者はおよそ 40% べき演算を削減している事が分かる. 両者を合わせると 48% 程度の削減となる. また通信データ量はおよそ 60% 削減している事が分かる. 同様に [プロトコル 7] は [プロトコル 5] と比べて証明者はおよそ 50%, 検証者はおよそ 33% べき演算を削減し, これは両者を合わせると 40% 程度の削減となり, 通信データ量はおよそ 75% 削減している事が分かる. 一方, [27] に [プロトコル 6] を適用した場合, 証明者及び検証者はそれぞれおよそ 44%, 32% べき演算を削減し, これは両者を合わせると 38% 程度の削減となる. 通信データ量は, $|g|=|p|$ とした場合, およそ 23% 削減している. 同様に [28] に [プロトコル 7] を適用した場合, 証明者及び検証者はそれぞれおよそ 40%, 25% べき演算を削減し, これは両者を合わせると 32% 程度の削減となる. 通信データ量は, $|g|=|p|$ とした場合, およそ 27% 削減している.

8. まとめと今後の課題

本稿では, 既存のマルチパーティ秘匿回路計算の計算・通信コスト削減に対して有効とおもわれる幾つかのバッチ処理を挙げ, その削減効果を検証した. その結果, Schoenmakers, Tuyls のマルチパーティ秘匿回路計算方式に対して, 計算・通信コストをそれぞれ 38% 及び 23% 程度削減出来る事が分かった. また我々のグループの方式に対しては, 計算・通信コストをそれぞれ 32% 及び 27% 程度削減出来る事が分かった. しかし本改良は, マルチパーティ秘匿回路計算プロトコルの実用化に対しては未だ十分とはいえない.

今後は本稿で紹介したバッチ処理プロトコルに対する厳密なセキュリティ考察, 及び実装による効果の検証を行うとともに, 更に別の観点から, 計算・通信コスト削減の可能性について考察する予定である.

文 献

[1] M. Abe, "Non-interactive and optimally resilient distributed multiplication," IEICE Trans. Fundamentals, Vol. E83-A, No. 4, pp. 598-605, April 2000.

[2] M. Abe, "Efficient components for cryptographic applications in the discrete-log setting," PhD thesis, University of Tokyo, 2002.

[3] D. Beaver, "Minimal-latency secure function evaluation," B. Preneel (Ed.), Advances in Cryptology — EUROCRYPT '00, LNCS 1807, pp. 335-350, Springer-Verlag, 2000.

[4] M. Bellare, J.A. Garay, and T. Rabin, "Fast batch verification for modular exponentiation and digital signatures," K. Nyberg (Ed.), Advances in Cryptology — EUROCRYPT '98, LNCS 1403, pp. 236-250, Springer-Verlag, 1998.

[5] M. Ben-Or, S. Goldwasser, and A. Widgerson, "Completeness theorems for non-cryptographic fault-tolerant distributed computation," In Proc. of the 20th Annual Symposium on Theory of Computing (STOC), pp. 1-10, ACM Press, 1988.

[6] E.F. Brickell and Y. Yacobi, "On privacy homomorphisms," D. Chaum and W.L. Price (Eds.), Advances in Cryptology — EUROCRYPT '87, LNCS 304, pp. 117-125, Springer-Verlag, 1988.

[7] E. Brickell, D. Gordon, K. McCurley, and D. Wilson, "Fast exponentiation with precomputation," R.A. Rueppel (Ed.), Advances in Cryptology — EUROCRYPT '92, LNCS 658, pp. 200-207, Springer-Verlag, 1992.

[8] D. Chaum, C. Crépeau, and I. Damgård, "Multiparty unconditionally secure protocols," In Proc. of the 20th Annual Symposium on Theory of Computing (STOC), pp. 11-19, ACM Press, 1988.

[9] 千田浩司, 谷口展郎, 山本剛, 岡崎崇人, 塩野入理, 金井敦, "エルガマル暗号に基づく秘匿回路計算の実装と応用," コンピュータセキュリティシンポジウム 2005 (CSS 2005) 論文集 Vol. II, pp. 475-480, (社) 情報処理学会, 2005.

[10] R. Cramer, I. Damgård, and B. Schoenmakers, "Proofs of partial knowledge and simplified design of witness hiding protocols," Y.G. Desmedt (Ed.), Advances in Cryptology — CRYPTO '94, LNCS 839, pp. 174-187, Springer-Verlag, 1994.

[11] R. Cramer, I. Damgård, and J.B. Nielsen, "Multiparty computation from threshold homomorphic encryption," Basic Research in Computer Science (BRICS) RS-00-14, June 2000.

[12] S. Even, O. Goldreich, and A. Lempel, "A randomized protocol for signing contracts," Communication of the ACM, Vol. 28, No. 6, pp. 637-647, ACM Press, 1985.

[13] P. Feldman, "A practical scheme for non-interactive verifiable secret sharing," In Proc. of the 28th IEEE Annual Symposium on Foundations of Computer Science (FOCS), pp. 427-437, IEEE Press, 1987.

[14] A. Fiat and A. Shamir, "How to prove yourself: Practical solutions to identification and signature problems," A.M. Odlyzko (Ed.), Advances in Cryptology — CRYPTO '86, LNCS 263, pp. 186-199, Springer-Verlag, 1987.

[15] M. Franklin and S. Haber, "Joint encryption and message-efficient secure computation," D.R. Stinson (Ed.), Advances in Cryptology — CRYPTO '93, LNCS 773, pp. 266-277,

Springer-Verlag, 1994.
[16] R. Gennaro, M. Rabin, and T. Rabin, "Simplified VSS and fast-track multiparty computations with applications to threshold cryptography," In Proc. of the 17th ACM Symposium on Principles of Distributed Computing (PODC '98), pp. 101–111, ACM Press, 1998.
[17] R. Gennaro, D. Leigh, R. Sundaram, and W. Yerazunis, "Batching Schnorr identification scheme with applications to privacy-preserving authorization and low-bandwidth communication devices," P.J. Lee (Ed.), Advances in Cryptology — ASIACRYPT 2004, LNCS 3329, pp. 276–292, Springer-Verlag, 2004.
[18] O. Goldreich, S. Micali, and A. Widgerson, "How to play any mental game, or a completeness theorem for protocols with honest majority," In Proc. of the 19th ACM Symposium on Theory of Computing (STOC), pp. 218–229, ACM Press, 1987.
[19] D. Malkhi, N. Nisan, B. Pinkas, and Y. Sella, "Fairplay – Secure two-party computation system," In Proc. of the 13th USENIX Security Symposium, pp. 287–302, 2004.
[20] P. Mohassel and M. Franklin, "Efficiency tradeoffs for malicious two-party computation," Y.G. Desmedt (Ed.), 9th International Workshop on Practice and Theory in Public Key Cryptography (PKC 2006), LNCS 2567, pp. 458–473, Springer-Verlag, 2006.
[21] 岡本龍明, 山本博資, 「現代暗号」, 産業図書, 1997.
[22] P. Paillier, "Public-key cryptosystems based on composite degree residuosity classes," J. Stern (Ed.), Advances in Cryptology — EUROCRYPT '99, LNCS 1592, pp. 223–238, Springer-Verlag, 1999.
[23] T.P. Pedersen, "Non-interactive and information-theoretic secure verifiable secret sharing," J. Feigenbaum (Ed.), Advances in Cryptology — CRYPTO '91, LNCS 576, pp. 129–140, Springer-Verlag, 1992.
[24] R. L. Rivest, L. Adleman, and M. L. Dertouzos, "On data banks and privacy homomorphisms," Foundations of Secure Computation, pp. 169–179, New-York: Academic Press, 1978.
[25] T. Sander, A. Young, and M. Yung, "Non-Interactive CryptoComputing for NC^1," In Proc. of the 40th IEEE Symposium on the Foundations of Computer Science (FOCS), pp. 554–567, IEEE Press, Oct. 1999.
[26] C.P. Schnorr, "Efficient signature generation for smart cards," Journal of Cryptology, Vol. 4, No. 3, pp. 239–252, 1991.
[27] B. Schoenmakers and P. Tuyls, "Practical Two-Party Computation Based on the Conditional Gate," P.J. Lee (Ed.), Advances in Cryptology — ASIACRYPT 2004, LNCS 3329, pp. 119–204, Springer-Verlag, 2004.
[28] G. Yamamoto, K. Chida, A. Nascimento, K. Suzuki, and S. Uchiyama, "Efficient, non-optimistic secure circuit evaluation based on the ElGamal encryption," J. Song, T. Kwon, and M. Yung (Eds.), Information Security Applications: 6th International Workshop (WISA 2005), LNCS 3786, pp. 328–342, Springer-Verlag, 2006.
[29] 山本剛, 千田浩司, "対話証明のバッチ実行," ISEC/SITE/IPSJ-CSEC 研究会予稿集, July, 2006.
[30] A.C. Yao, "Protocols for secure computation," In Proc of the 23rd IEEE Symposium on Foundations of Computer Science (FOCS), pp. 160–164, IEEE Press, 1982.

付　　録

1. プロトコル 8

[プロトコル 8]
P の秘密入力: $(b_i, x_{i,b_i}) \in \{0,1\} \times \mathbb{Z}/p\mathbb{Z}$
共通入力: $\mathcal{G}$, p, g, $(y_{i,0}, y_{i,1}) = (g^{x_{i,0}}, g^{x_{i,1}}) \in \mathcal{G}^2$ $(i = 1, \ldots, d)$

(1) P は $r, v, c_{i,1-b_i} \in_R \mathbb{Z}/p\mathbb{Z}$ を選び $(i = 1, \ldots, d)$,
$R_0 = g^r \prod_{\{i|b_i=1\}} y_{i,0}^{c_{i,0}}$, $R_1 = g^v \prod_{\{i|b_i=0\}} y_{i,1}^{c_{i,1}}$,
を計算し, $i = 1, \ldots, d$ について順に
$c_i = \mathcal{H}(c_{i-1} \| c_{i-1,0})$, $c_{i,b_i} = c_i - c_{i,1-b_i} \bmod p$,
を計算し (但し $c_0 = R_0$, $c_{0,0} = R_1$), その後
$z_0 = r - \sum_{\{i|b_i=0\}} c_{i,0} x_{i,0} \bmod p$,
$z_1 = v - \sum_{\{i|b_i=1\}} c_{i,1} x_{i,1} \bmod p$,
を計算して $(c_1, c_{1,0}, \ldots, c_{d,0}, z_0, z_1)$ を V に送信する.

(2) V は $i = 1, \ldots, d$ について順に
$c_{i,1} = c_i - c_{i,0} \bmod p$, $c_{i+1} = \mathcal{H}(c_i \| c_{i,0})$,
を計算し (但し c_{d+1} は計算不要),
$c_1 = \mathcal{H}(g^{z_0} \prod_i y_{i,0}^{c_{i,0}} \| g^{z_1} \prod_i y_{i,1}^{c_{i,1}})$,
が成り立てば P の証明を受理する.