解　説

# ゼロ知識証明とマルチパーティプロトコル†

黒沢　馨††　岡本龍明†††

## 1. まえがき

「Alice と Bob は離婚したばかりで現在別々の町に住んでいる．彼らが所有していた車をどちらが引き取るか，電話でコイン投げ（じゃんけん）をして決めたい．では，そのコイン投げをどのように実現すればよいであろうか？」

このような書き出しで始まる Blum の 1982 年の論文[1]が，暗号を用いて多様な機能を実現するマルチパーティプロトコル (Multi-party protocol) の始まりの一つとなり，さらにゼロ知識証明の始まりでもあった．つまり，現代の暗号理論における最も華やかな成果であるゼロ知識証明とマルチパーティプロトコルは，ほぼ誕生を同じくし，その誕生の時点ですでに，後にみられる相互の関係（マルチパーティプロトコルを安全に実現するためにゼロ知識証明が利用される）が明確に実現されていたのである．

1980 年代前半には，上記コイン投げプロトコルのほかに，メンタルポーカプロトコル（電話で，公平にポーカを行う方法)[25]，金持ちの財産比べプロトコル（お互いに自分の財産を秘密にして，どちらが金持ちかを知る方法)[26]，同時文書交換プロトコル（電話で同時に情報を交換する方法)[33] などの暗号を利用した多彩なプロトコルが数多く発表された．これら各プロトコルは，一見不可能にみえる個々の問題を巧みに解決している具体例として大変興味深いものである．その後，1986 年に Yao により，このような個々のプロトコルをすべて包括する一般化が行われ，それが現在われわれがマルチパーティプロトコルと名付けている研究の始まりとなった[34]（ただし，この Yao の論文では，2 者間のプロトコルに限定されている）．

一方，本小特集号のテーマであるゼロ知識証明は，先に述べた Blum の先駆的仕事から約 3 年後に，Goldwasser らにより見事に定式化されたことは，本号の小山氏の記事などで紹介されているとおりである．その後，ゼロ知識証明は，暗号を応用した種々のプロトコルを安全に構成する上で，必要不可欠の道具として多用されていく．つまり，上記 Blum の論文にみられるように，ゼロ知識証明は，元来，プロトコルを安全に構成するための道具として誕生したと考えられ，それがその後，多くのプロトコルを構成する上で，必要不可欠な道具となることは，必然的結果とも言える．Goldwasser たちによるゼロ知識証明の発表 (1985 年) の直後に，Benaloh (Cohen) は選挙（無記名投票）のプロトコルにゼロ知識証明を応用し[3),4)]，1986 年には，Crépeau はメンタルポーカにゼロ知識証明を応用している[5]．（なお，Fiat and Shamir は，ゼロ知識証明を応用した有名な認証法をほぼ同時期に発表している[6]．この方面への応用については，本号の太田，藤岡氏の解説記事を参照されたい．）そして現在，ゼロ知識証明は，これらコイン投げ，選挙，メンタルポーカなどの個々のプロトコルに限らず，先ほど述べた一般のマルチパーティプロトコルを構築する上で本質的に不可欠なツールとして利用されているのである[7)~13)]．

本稿では，このマルチパーティプロトコルとはどういうものか，その中でゼロ知識証明がどのように使われているのかといったことを中心に，マルチパーティプロトコルの紹介を行う．マルチパーティプロトコルは，ゼロ知識証明を始め現在までに暗号の世界で開発された種々のテクニックが多用されるため，その全貌を簡単に紹介することは，非才な著者らにとっては大変難しい仕事である．ただ，ここでご紹介するいくつかの例を通

† Zero-Knowledge Proofs and Multi-Party Protocols by Kaoru KUROSAWA (Tokyo Institute of Technology, Department of Electrical and Electronic Engineerings) and Tatsuaki OKAMOTO (NTT Communications and Information Processing Loboratories).

†† 東京工業大学工学部電気電子工学科
††† NTT 情報通信処理研究所

して，マルチパーティプロトコルが扱う世界の多彩さの一端でも感じていただければ幸いである．

本稿は，以下のように，構成される．まず，**2.** でマルチパーティプロトコルとは，どういうものであるかを概説する．次に，**3.** で，マルチパーティプロトコルを始めとする多くの暗号応用プロトコルにおいて基本的ツールとして大変よく用いられる，Verifiable Secret Sharing (VSS) と Oblivious Transfer (OT) を紹介する．なお，VSS では，すでにゼロ知識証明がその構成要素として使われることになる．**4.** で，マルチパーティプロトコルの最も典型的実現例である選挙（無記名投票）の実現例とその中でのゼロ知識証明の利用方法を述べる．最後に，**5.** で一般的マルチパーティプロトコルの実現例をゼロ知識証明との関連を中心に述べる．

## 2. マルチパーティプロトコルとは

たとえば，無記名投票による信任選挙を考えよう．ここでは，各投票者は，信任か不信任かを無記名で投票用紙に記入する．最後に，集計者が，信任と投票した人の数 $T$ を公開する．ここで，投票者は $n$ 人いるとし，各投票者 $i$ $(i=1,\cdots,n)$ の投票内容を1ビットの情報 $x_i$（信任ならば1とし，不信任ならば0とする）で表すと，投票の集計結果 $T$ は，$T=x_1+\cdots+x_n$ と書ける．この選挙システムにおいては，だれが信任としたか不信任としたかは秘密にしたまま，何人信任したかという情報のみを知ることができる．つまり，このような選挙の問題は，各投票者 $i$ の保持する情報 $x_i$（0か1）を秘密にしたまま，それらの総和 $T=x_1+\cdots+x_n$ を計算する問題と考えることができる．

さらに，この問題は，次のような問題に一般化される．今，$n$ 人の参加者より構成されるネットワークがあるとして，それぞれの参加者 $i$ が自分だけが知っている秘密の情報 $x_i$（たとえば，投票する内容）を保持しているとする．ある関数 $f$（たとえば，足し算）が与えられたとき，各参加者は，各自の秘密情報を秘密にしたまま $y=f(x_1,\cdots,x_n)$（たとえば，$y=x_1+\cdots+x_n$）を計算しようとする．各参加者は，ネットワーク上で，各自の秘密を一切漏らさないで，お互いになんらかの通信を行い，最終的に全員が $y$ の値を正しく知ることができるような機能を実現する問題を考える．このような機能を実現するプロトコルを関数 $f$ に対するマルチパーティプロトコルという（つまり，総和関数に対するマルチパーティプロトコルは，上記のような選挙を実現するプロトコルとなる）．

もし，信頼できるセンタ（たとえば，信頼できる開票者）と秘密通信路を仮定すると，各参加者 $i$ は，秘密通信路を用いて $x_i$ をセンタに送り，センタは，$y=f(x_1,\cdots,x_n)$ を計算し，$y$ を各参加者に送信することで，この機能を容易に実現できる．しかし，信頼できるセンタという仮定は，センタ運用などの非技術的要因に依存したきわめて大きな仮定であるため，一般にマルチパーティプロトコル問題では，単に秘密通信路の存在（慣用暗号の存在）や，公開鍵暗号の存在といったより技術的に妥当な仮定のみを前提に構成するものとする．さて，$n$ 人の参加者の中には，何人かの不正者がいて，プロトコル実行中に，本来のプロトコルから逸脱した行為を行い，他人の秘密（たとえば，他人の投票内容）を盗もうとしたり，正しい計算結果（たとえば，正しい選挙結果）が出ないように攪乱しようとするような不正行為を行うことが考えられる．このような不正者が $t$ 人いるとして，あらゆる（$t$ 人の）不正行為を考えても，彼らが他人の秘密を盗むことも，計算結果を攪乱することもできないようなマルチパーティプロトコルがあれば，このようなプロトコルを $t$-安全であるという．

一方，マルチパーティプロトコルは，各参加者 $i$ が自分の手の内 $x_i$ を秘密にしたまま，あるゲームにより勝敗を決めるような場合にも利用できる．ここでは，関数 $f$ は，ルールに従い勝敗（もしくは順位など）を定める関数になる．つまり，任意の関数に対するマルチパーティプロトコルを構成することにより，ポーカゲームなどの任意のメンタルゲームを実行するプロトコルが構成できることにもなる．

## 3. マルチパーティプロトコルで利用される基本ツール

本章では，マルチパーティプロトコルで，基本ツールとして用いられる，Verifiable Secret Sharing (VSS) と Oblivious Transfer (OT) を紹介する．

### 3.1 Verifiable Secret Sharing (VSS)

マルチパーティプロトコルでは，各参加者の秘密情報を分割して他の参加者に配布し，分割されたままで計算を行ったり，必要に応じて元の情報に復元したりすることが必要となる．このような目的にかなうツールが，VSS である[27]．

VSS は，Shamir の Secret Sharing (SS) をゼロ知識証明と組み合わせることにより構成されるものである．まず，SS の紹介を行う．分配者が，ある秘密を $n$ 個の分割情報に分割し，それぞれを $n$ 人の分割保持者に配る．そのうち $k$ 人の分割保持者が集まると，元の秘密が復元できる方式を $(k, n)$ しきい値 Secret Sharing (SS) という．Shamir は，多項式補間を利用することにより，以下のような SS の実現法を提案している．

(1) 分配者は，自分の秘密 $m$ を定数項とする randam な $(k-1)$ 次の多項式 $f(x)$ を選ぶ．

$$f(x)=m+a_1x+a_2x^2+\cdots+a_{k-1}x^{k-1} \bmod r$$

分配者は，各分割保持者 $j$ $(j=1,\cdots,n)$ に $f(j)$ を配る．(ここで，$r$ は，$m<r$ となるような適当な素数とする．)

(2) $n$ 個に分割された分割情報 $f(j)(j=1,\cdots,n)$ の中で，いかなる $(k-1)$ 個の分割情報を用いても $m$ を復元することはできないが，$k$ 個以上の分割情報を用いれば，必ず $m$ を復元できる．

さて，次に，SS を用いて VSS を構成する方法を述べる．上記の SS では，分配者が手順通りに分割情報を作らないと，$k$ 個以上の分割情報を用いても秘密を正しく復元できないことがある．そこで，VSS では，分配者が，正しく手順通りに分割情報を作っていること (つまり，どのような $k$ 人以上の分配者により秘密を復元しても，同じ秘密が復元されるようになる) をゼロ知識証明で証明する．VSS の手順は，以下のとおりである．

(1) 秘密 $m$ をもつ分配者は，$m$ を $c(m)$ に暗号化し，それを $n$ 人の分割保持者に送る．

(2) 分配者は，SS を用いて，各分割保持者 $j$ $(j=1,\cdots,n)$ に $f(j)$ を配る．

(3) 分配者は，各分割情報が上記に述べた手順により正しく作成されていることを各分割保持者に対しゼロ知識証明により証明する．なお，各分割情報が正しく作成されているかどうかという問題は，NP 問題であるため，一方向性関数が存在すれば，それを証明するゼロ知識証明は必ず構成できる (静谷，伊東，桜井氏らの解説記事を参照)．

なお，**4.6** において，多次剰余暗号系を通信路として用いた場合における，VSS の効率的具体的構成例を示す．

### 3.2 Oblivious Transfer (OT)

マルチパーティプロトコルでは，秘密を保証しながらそれに関連するある情報を送りたいような状況が生じる．そのような状況では，Oblivious Transfer が，大変有効なツールとして用いられる．Oblivious Transfer は，いくつかの種類があるが，その典型例は，1-out-of-2 Oblivious Transfer (1-2 OT) である．これは，送り手が，二つの情報 $m_1$ と $m_2$ を保持しており，そのうちのいずれか一つのみを受け手に送るが，受け手がどちらを受け取ったかを送り手は知ることができないようなものである．たとえば，二つの情報を別々に郵便で送り，いずれか一方が，必ず途中で紛失し，一つのみが無事受け手に送られるような場合である．このとき，送り手は，いずれの情報が無事届いたかを知ることはできない．一方，受け手は，届かなかったほうの情報については，一切知ることができない．その実現法としては，ここでは記さないが，公開鍵暗号があれば，OT が実現できることが文献 33) で示されている．OT が一般のマルチパーティプロトコルにおいてどのように用いられるかは，**5.** で触れるが，その他の応用として，同時文書交換プロトコルなどへの応用が文献 33) で紹介されている．

## 4. 選挙プロトコル—マルチパーティプロトコルの典型例—

本章では，分かりやすいマルチパーティプロトコルの一例として，選挙のプロトコルを示す．センタが一つ存在する方式からセンタなしの方式までの流れを通じ，マルチパーティプロトコルの考え方，雰囲気などを感じとっていただきたい．

### 4.1 多次剰余暗号系

これは，準同型性を有する多値確率的暗号系である．多値数が $r$ の場合，以下のような暗号系となる．

(秘密鍵) 二つの大きな素数 $p$ と $q$

(公開鍵) $N(=pq)$, $y$

(平文) $m$ $(0\leq m<r)$

(暗号文) $E(m)=y^m x^r \bmod N$，ただし $x$ は乱数．

次式が成り立つことは，容易に分かる．

$$E(m+n)=E(m)E(n)x^r \bmod N, {}^{\exists}x$$

この準同型性が，選挙などにおいて，重要な役割を果たす．

[定理][12),13)]

上記の暗号系において，$Z_N^*=\{x|0<x<N, gcd(x,N)=1\}$ の任意の要素が一意に復号できるための必要十分条件は，以下の式(1)～(6)が成り立つことである．

$$y^j \neq x^r \bmod r \text{ for any x } (1\leq j<r) \quad (1)$$

$$gcd(p-1,r)=e_1 \quad (2)$$

$$gcd(q-1,r)=e_2 \quad (3)$$

$$r=\begin{cases} e_1e_2 & \text{if } r \text{ is odd.} \\ (e_1e_2)/2 & \text{if } r \text{ is even.} \end{cases} \quad (4)$$

$$gcd(e_1,e_2)=\begin{cases} 1 & \text{if } r \text{ is odd.} \\ 2 & \text{if } r \text{ is even.} \end{cases} \quad (5)$$

$$(y/N)=1 \text{ if } r \text{ is even.} \quad (6)$$

(復号法)

mod $p$ において，

$$\{E(m)\}^{(p-1)/e_1}=(y^mx^r)^{(p-1)/e_1}$$
$$=(y^{(p-1)/e_1})^m(x^{r/e_1})^{(p-1)}$$
$$=(y^{(p-1)/e_1})^m$$

同様に，mod $q$ において，

$$\{E(m)\}^{(q-1)/e_2}=(y^{(q-1)/e_2})^m \bmod q$$

したがって，$1\leq i<r$ になる $i$ について，

$$\{E(m)\}^{(p-1)/e_1} \bmod p \text{ and } \{E(m)\}^{(q-1)/e_2} \bmod q.$$

と

$$(y^{(p-1)/e_1})^i \bmod p \text{ and } (y^{(q-1)/e_2})^i \bmod q.$$

を比較すればよい．

### 4.2 センタが一つの方式[3)]

不正なセンタは，

(1) 偽りの投票結果を公表するかもしれない．

(2) ある投票者には，2票以上の投票権を与えるかもしれない．

本節では，以上のような不正が不可能な方式を示す．

一般に，マルチパーティプロトコル設計の基本思想は，以下のとおりである．まず，各参加者がルール違反をしないかぎり，プライバシが保たれるプロトコルを設計する（基本プロトコル）．しかるのち，各参加者はこのプロトコルに従って暗号文を送信するたびに，ルール違反をしていないことをゼロ知識対話型証明で示す(verification)．

選挙の基本プロトコルは，以下のようになる．

(1) センタは，$r$ 値の確率的暗号系を構成し，公開鍵パラメータ $(N,y)$ を公開する．ただし，投票者数 $h$ は $r$ 未満とする．

(2) 投票者 $i$ は，自分の投票 $m_i(=0 \text{ or } 1)$ をセンタの公開鍵で暗号化し，それを

$$z_i=y^{m_i}x_i^r \bmod N$$

として公開する．ただし，$x_i$ は乱数．

(3) センタは，それらを復号し，

$$M=m_1+m_2+\cdots m_h$$

を投票結果として公開する．

各ステップに対応する verification は，以下のようになる．

(1) センタは，$(N,y)$ が $r$ 値の確率的暗号系の公開鍵パラメータの条件を満たしていることをゼロ知識対話型証明で示す[12)]．

(2) 各投票者 $i$ は，$z_i$ の平文が $m_i=0$ or 1 を満たしていることをゼロ知識対話型証明で示す(次節参照)．

(3) センタは，

$$z_1z_2\cdots z_h=y^Mx^r \bmod N$$

が成り立っていることを，ゼロ知識対話型証明で示す．(多次剰余暗号系の準同型性より，上式が成り立つ．) つまり，

$$z_1z_2\cdots z_h/y^M=x^r \bmod N$$

を示せばよい．

多次剰余暗号系の条件から，上式は複数の解 $x$ を有する．したがって，センタが $x$ を公開し，かつ，全投票者が結託すると，$N$ が素因数分解できてしまう．(全投票者が結託することはない，と仮定すれば，センタは単に $x$ を公開するだけでよい．)

### 4.3 暗号カプセル[9)]

前節の(2)の verification，つまり，

"$z_i$ の平文は，0 or 1"

のゼロ知識対話型証明を示す．

(step 1) Aは，

$$z=y^mx^r \bmod N$$

をBに送る．ただし，$m=0$ or 1.

以下の step 2～5 を $i=1\cdots t$ について，繰り返す．ただし，$t$ は $N$ のビット数．

(step 2) Aは，乱数 $s_i$，$t_i$ を選び，

$$u_i=s_i^r \bmod N$$
$$v_i=yt_i^r \bmod N$$

を計算する．この $u_i, v_i$ をランダムな順番でBに送る．

(step 3) Bは，ランダムに $e_i=0$ or 1 をAに送る．

(step 4) Aは，

$e_i=0$ であれば，$s_i$, $t_i$ をBに送る．

$e_i=1$ であれば，$u_i$, $v_i$ のうち，$z$ と等価な（平文が同じ）ほうを $w$ とする．すると，

$$z/w=x_i^r \bmod N \text{ for some } x_i$$

となるので，この $x_i$ をBに送る．

(step 5) Bは，送られてきたものをチェックする．

上記のいずれかが成り立たない場合，プロトコルは，その時点で停止する．

一般に，$(u_i, v_i)$ のように，ランダムな順番に並べられた，特定の仕様を満たすオブジェクトの組を暗号カプセルという．暗号カプセルは，平方非剰余，グラフ非同型，SAT などのゼロ知識対話型証明にも，広く，応用されている．

### 4.4 センタが複数の方式[4]

**4.2** の方式では，センタに各投票者の投票内容が分かってしまう．センタを複数にすると（$n$ 個），この欠点を取り除くことができる．

基本プロトコルは，以下のようになる．

（1） センタ $j$ は，$r$ 値の確率的暗号系を構成し，公開鍵パラメータ $(N_i, y_i)$ を公開する．ただし，投票者数 $h$ は $r$ 未満とする．

（2） 投票者 $i$ は，自分の投票 $m_i$ (=0 or 1) を

$$m_i=m_{i1}+m_{i2}+\cdots+m_{in} \bmod r$$

と $n$ 個に分割する．$m_{ij}$ をセンタ $j$ の公開鍵で暗号化し，それを

$$z_{ij}=y_j^{m_{ij}}x_{ij}^r \ b \ mod \ N_j$$

として公開する．ただし，$x_{ij}$ は乱数．

（3） 各センタ $j$ は，自分に送られてきたものを復号し，

$$M_j=m_{1j}+m_{2j}+\cdots m_{hj} \bmod r$$

を公開する．投票結果は，

$$M=M_1+M_2+\cdots M_n \bmod r$$

として与えられる．

Verification については，省略する．各自，構成してみることをおすすめする．

### 4.5 不正なセンタを許す方式[23]

前節の方式では，センタが一つでも故障すると，選挙は失敗する．センタのゼロ知識証明が不成立の場合も，選挙は失敗する．前に述べた，VSS を利用すると，この問題を解決できる．すなわち，$(n-1)/2$ 個までの，故障，結託，その他プロトコルから逸脱する不正なセンタが許容されるのである．まず，本節では，SS を用いたプロトコルを示し，次節で，その VSS の具体的構成法を示す．

（1） センタ $j$ は，$r$ 値の確率的暗号系を構成し，公開鍵パラメータ $(N_j, y_j)$ を公開する．ただし，投票者数 $h$ は $r$ 未満とする．

（2） 投票者 $i$ は，自分の投票 $m_i$ (=0 or 1) を定数項とする random な $(k-1)$ 次の多項式 $f_i(x)$ を選ぶ（係数は mod $r$）．

$$f_i(x)=m_i+a_1x+a_2x^2+\cdots+a_{k-1}x^{k-1}$$

$f_i(j)$ をセンタ $j$ の公開鍵で暗号化し，それを公開する．

（3） 各センタ $j$ は，自分に送られてきたものを複号し，

$$M_j=f_1(j)+f_2(j)+\cdots f_h(j) \bmod r$$

を公開する．ここで，

$$F(x)=f_1(x)+f_2(x)+\cdots f_h(x)$$

は，$m_1+m_2+\cdots m_h$ を定数項とする $(k-1)$ 次の多項式になっている．したがって，$(n-k)$ 個のセンタが故障したとしても，$M_j$ が $k$ 個分かれば投票結果を復元できる．

### 4.6 Verifiable secret sharing[23]

**4.5** のプロトコルでは，**3.1** で紹介した SS 及び VSS を利用する．本節では，多次剰余暗号系を利用した VSS の一構成法を示す．

以下では，**3.1** と同様の記号を用いる．不正な分配者は，乱数を $f(j)$ として配るかもしれない．すると，1番から $k$ 番までの分割保持者が復元して得られる結果と，2番から $k+1$ 番までの分割保持者が復元して得られる結果が異なってしまう．これを防ぐには，分配者がプロトコルに従っていることをゼロ知識対話型証明で示せばよい．これが，VSS である．

$i$ 番の分割保持者の $r$ 次剰余暗号系の公開鍵を $(N_i, y_i)$ とする．

(step 1) 分配者は，$s_i=f(i)$ を，$i$ 番の分割保持者の多次剰余暗号系で暗号化し，

$$z_i=y_i^{s_i}x_i^r \bmod N_i$$

を公開する．$i$ 番の分割保持者は，これを復号し，自分の share $s_i$ を得る．

以下は，

$L=\{z_1,\cdots z_n | z_i=y_i^{s_i}x_i^r \bmod N_i, s_i=f(i)\}$

に対するゼロ知識対話型証明である．

以下の step 2〜4 を $t$ 回繰り返す．ただし，$t$ は $N$ のビット数．

(step 2)　分配者は，ランダムな $k-1$ 次の多項式 $f'$ を選ぶ．$f'$ について，step 1 と同様なことを行う．すなわち，分配者は，$s_i'=f'(i)$ を，$i$ 番の分割保持者の多次剰余暗号系で暗号化し，

$z_i'=y_i^{s_i'}x_i'^r \bmod N_i$

を公開する．

(step 3)　分割保持者は，ランダムに $e=0$ or 1 を分配者に送る．（$n$ 人の分割保持者が，なんらかの合意の上，ランダムに $e=0$ or 1 を分配者に送る．）

(step 4)

$e=0$ の場合，分配者は，$s_i'$ と $x_i'$ をすべてオープンし，$f'$ が高々 $d$ 次の多項式であることを示す．

If $e=1$ の場合，分配者は，

$z_iz_i'=y_i^{t_i}w_i^r \bmod N_i$

を満たす $t_i$ と $w_i$ をすべてオープンし，$f+f'$ が高々 $d$ 次の多項式であることを示す．

### 4.7　暗号文を公開しない方式[11]

前節までの方式においては，各投票内容の暗号文が公開される．本節では，各参加者間の安全な秘密通信路を仮定し，暗号文は公開しない，という方式を示す．これは，$(k,n)$ しきい値法が BCH 符号[24]になっている[23]ことを利用している．すなわち，

（1）　$r$ を素数，

（2）　$w$ を mod $r$ における原始 $n$ 乗根，

（3）　$f$ を $t$ 次の任意の多項式（係数は mod $r$）

（4）　$n=3t+1$ とすると，

$f(1), f(w), \cdots f(w^n)$

は，$t$ 重誤り訂正 BCH 符号である．

そこで，センタの個数 $n$ は $n=3t+1$ とし，不正なセンタの数は $t$ 以下とする．

基本プロトコルは，以下のようになる．

（1）　投票者 $i$ は，自分の投票 $m_i(=0$ or $1)$ を定数項とする random な $t$ 次の多項式 $f_i(x)$ を選ぶ（係数は mod $r$）．$f_i(w^j)$ を秘密通信路によってセンタ $j$ に送る．

（2）　各センタ $j$ は，自分に送られてきたものを復号し，

$M_j=f_1(w^j)+f_2(w^j)+\cdots f_h(w^j) \bmod r$

を公開する．ここで，

$f_1(x)+f_2(x)+\cdots f_h(x)$

は，$m_1+m_2+\cdots m_h$ を定数項とする $t$ 次の多項式になっている．したがって，$t$ 個のセンタが不正をしたとしても，BCH 符号の誤り訂正能力によって投票結果を復元できる．

### 4.8　センタなしの方式

センタが複数の方式において，投票者がセンタを兼ねれば，センタが不用な方式となる．これは，足し算のマルチパーティプロトコルと呼ばれる．任意のブール関数に対する（センタが不用な）多者プロトコルは，次章以降で解説する．

## 5. 一般のマルチパーティプロトコル

前章では，マルチパーティプロトコルの一実現例として，選挙（無記名投票）を実現するプロトコルを示した．本章では，任意の関数に対してセンタを必要としない一般的なマルチパーティプロトコルをゼロ知識証明との関わりを中心に簡単に紹介する．

### 5.1　研究の動向

**1.** で述べたように，マルチパーティプロトコルは，1980 年前後に発表された暗号機能を利用したポーカゲームプロトコル[25]，金持ちの財産比べプロトコル[26]，コイン投げプロトコル[1]などをルーツとして生まれたものである．この研究が現在のような形で，$n$ 人のマルチパーティプロトコルとして，その実現法が発見されるためには，Yao の 2 者プロトコル[34]，ゼロ知識証明（1985 年文献 2））及びその応用である Verifiable Secret Sharing（VSS, 1985 年，文献 27））の発表を待つ必要があった．

結局，1987 年に，Yao のアイデア，ゼロ知識証明及び VSS を利用することにより，任意の関数に対する最初のマルチパーティプロトコルが発表された[7]．ここでは，“公開鍵暗号（落とし戸付き一方向置換）の存在を前提にして，任意の関数 $f$ 及び $t<n/2$ に対して，$t$-安全なマルチパーティプロトコルが存在する”という結果が得られている（この結果は，**4.** で述べた選挙の例でみれば，**4.5** の方式のセンタなし版に対応する．）この論文を契機として，マルチパーティプロトコルの研

究は大変活発に行われてきた（極言すれば，1985年以降の理論的暗号研究の二大中心テーマが，ゼロ知識証明とマルチパーティプロトコルであったと言っても言い過ぎでないだろう）．さて，これらマルチパーティプロトコルの研究は，以下のような疑問に答える形で進展してきたと言える．

● 文献7）では，“公開鍵暗号の存在”が前提となっているが，この前提をもっと弱めた“秘密通信路の存在”や“放送型通信路（全参加者に同じ情報を同時に配送する通信路）の存在”を前提としてマルチパーティプロトコルを構成するにはどうすればよいか？ ここで，公開鍵暗号の存在というような仮定は，一方向性関数の存在といったいわゆる“計算量的仮定”を必要とするため，無限の計算能力をもつ不正者に対してはその仮定は成立しない（つまり，そのような仮定の下でのプロトコルは，無限の計算能力をもつ不正者に対しては安全でない）．一方，秘密通信路の存在というような仮定は，計算量的仮定ではなく情報理論的仮定であり，無限の計算能力をもつ不正者に対しても成立する（つまり，無限の計算能力をもつ不正者に対しても安全である）．安全性が，このような意味で保証されているとき，“無条件に安全である”ということがある．

● $t$-安全であるような $t$ の値を高めるにはどうすればよいか？ また，その理論的上限値（下界）はどのような値か？

● マルチパーティプロトコルを実行するのに必要なやりとりの回数（ラウンド数）はどれくらいか？

まず，1988年に次の結果が発表された[11),28)]．“秘密通信路の存在を前提にして，任意の関数 $f$ 及び $t<n/3$ に対して，$t$-安全なマルチパーティプロトコルが存在する．一方，この前提の下では，$t\geq n/3$ であるような $t$-安全なプロトコルは存在しない”．（この結果は，**4.** で述べた選挙の例でみれば，**4.7** の方式のセンタなし版に対応する）．

引き続き，1989年に，$t$ の上限値を $t<n/2$ に向上させた次のような結果が発表された[29)]．“秘密通信路と放送型通信路の存在を前提にして，任意の関数 $f$ 及び $t<n/2$ に対して，$t$-安全なマルチパーティプロトコルが存在する”．ここでは，放送型通信路の存在という仮定が新たに加わっているが，この仮定も依然，計算量的仮定ではない．

さて，このように，計算量的仮定を前提としない条件下では，$t$ の上限値が $n/2$ を越えることは，ほぼ不可能のように思われる．そこで，計算量的仮定の下で，$t$ の上限値を高める研究も行われている．1989年には，次のような結果が発表されている[30)]．“Oblivious Transfer の存在を前提にして，任意の関数 $f$ 及び $t<n$ に対して，$t$-安全なマルチパーティプロトコルが存在する”．なお，**3.2** で述べたように，Oblivious Transfer は，公開鍵暗号を用いて構成できることが知られている[33)]．

一方，プロトコルを効率の観点より眺めると，プロトコル終了時までに，何回ぐらい情報のやりとりがあったか（ラウンド数）ということが問題となる．上記で述べた方式では，このラウンド数は，いずれも与えられた問題（関数）のサイズに関して，多項式のオーダとなる（つまり，サイズの大きな問題を扱うとラウンド数がそれに応じて大きくなる）．ところが，1990年には，問題のサイズに依存しない（コンスタンスな）やりとりの回数でマルチパーティプロトコルを実現する方式[31),32)]が報告されている．

### 5.2 マルチパーティプロトコルとゼロ知識証明

マルチパーティプロトコルを構成するために，ゼロ知識証明がどのように使われるかは，**4.** の選挙のプロトコルの実現例で端的に示されているが，ここでは，一般のマルチパーティプロトコルの枠組みの中で，文献7），11）の構成を簡単に説明する．

いずれの方法においても（**4.** でも示したように），まず最初に，$n$ 人の全参加者が不正をしないという前提の下で，マルチパーティプロトコルを構成し（基本プロトコル），そののち，基本プロトコルとゼロ知識証明を組み合わせ，$t$ 人の不正者がいても安全性が保証されるプロトコルを構成する．なお，**5.2.1** の方式では，各参加者間の通信は，すべて公開鍵暗号を利用して行うものとし，**5.2.2** の方式では，各参加者間の通信は，秘密通信路を用いて行うものとする．

#### 5.2.1 公開鍵暗号を前提とする方式[7)]

**4.** の選挙プロトコルは，関数 $f$ が足し算である場合の特殊構成例であるが，ここでは，任意の

関数$f$に対して構成できることを示す．任意の関数$f$に対しては，$f$を計算するブール回路が構成できるため，この回路でのブール演算のマルチパーティプロトコルの実現法を考える．このような，回路は，通常ANDゲート，ORゲート，NOTゲートを組み合わせて構成されるが，ORは，ANDとNOTで実現できるため，ANDとNOTで構成されているとする．

（1） 基本プロトコル

さて，最初に，基本プロトコルは，以下のように構成される．まず，各参加者$i$は，秘密情報$x_i$を以下の関係を満たす$n$個のランダムな情報$b_1, \cdots, b_n$に分割し，$b_j$を他の利用者$j$ ($j=1, \cdots, n$)に分配する．

$$x_i = b_1 \oplus \cdots \oplus b_n$$

さて，これ以降の計算は，すべて秘密が$n$人の間で分割されたままで行われる．したがって，いかなる計算途中段階の情報も分割された状態であり，$n$人全員が協力しないかぎり情報を復元できない．最後に，$f$の計算が終了した段階で，全員が各自の最後の計算結果を公開し，それら分散された計算結果情報を復元することにより，最終計算結果のみを全員が知ることができる．さて，情報を分割したまま，任意の回路の計算ができることを示すためには，AND及びNOTに対しそのような計算ができることを示せば良い．ここで，その詳細を示すことは紙数の関係でできないが，NOTは各自が個別に計算でき，ANDは，参加者間でのOblivious Transfer (OT)を用いた通信に基づき実現できる（**3.2**で述べたように，OTは，公開鍵暗号を用いて構成できる）．なお，この基本プロトコルは，**4.**の例では，**4.4**の方式のセンタなし版に対応する．

（2） 半数以内の不正者を許すプロトコル

このようにして，全参加者が，（表面上）不正をしないという前提で，基本プロトコルが構成されるが，次に，$t$人の不正者がいても安全であるようなプロトコルを構成する．まず，各参加者は，各自の秘密をVSSを用いて$n$個に分割しそれぞれを他の参加者に分配する．ここで，このVSSの秘密を復元できるしきい値を$t<n/2$とする．また同時に，参加者$i$がこれ以降のプロトコルで用いる乱数$r_i$を$r=r_{i1} \oplus \cdots \oplus r_{in}$に分割し，$r_{ij}$を参加者$j$に送り，各$r_{ij}$をさらにVSSで$n$個に分割してそれぞれを各参加者に分配する．さらに，上記手順を正しく行ったこと（これは，NP問題になる）をゼロ知識証明で行う．以上の準備の後，各参加者は，基本プロトコルを実行するが，その際に，情報を送るごとに，各参加者は，正しくプロトコルに従っていることをゼロ知識証明で示す．もう少し詳しく示すと，ある情報$m$を受け手の公開鍵暗号で暗号化して$M=E(m)$を送る際には，それに関する以下のような命題をゼロ知識証明で証明する．「事前に行った手順で作った乱数$r_i$及び，正しい入力情報を用いて，正しい手順に従って作られた$m$を暗号化したものが$M$である．」以上のようなやり方で，基本手順を順次実行していくが，もし，ゼロ知識証明で失敗するようなことが起こった場合（つまり，だれかが不正をした場合），$n-t$人の正当な参加者が，VSSで分割した情報を持ち寄って，不正をした参加者$j$の秘密入力情報$x_j$及び乱数$r_j$を復元し，これ以降の参加者$j$の実行を正当な参加者で代行する（$x_j$と$r_j$があれば，だれでも参加者$j$を代行できる）．VSSのしきい値が$t<n/2$であるから，上記方式は，$t$-安全なマルチパーティプロトコルとなっている．なお，このプロトコルは，**4.**の例では，**4.5**の方式のセンタなし版に対応するものであるが，**4.5**の方式は，基本方式の部分をVSS機能と重畳させた簡略版である．

### 5.2.2 秘密通信路を前提とする方式[11]

まず，前の方法と同様に，不正者がいないことを前提に基本プロトコルを構成し，その後，$t$人の不正者に対して安全な方法に変換する．

（1） 基本プロトコル

この基本プロトコルでは，Shamirの Secret Sharing (SS)が用いられる．（前に示した方式では，Oblivious Transfer (OT)が必要であり，本方式では，公開鍵暗号系を前提としておらずそれを使えないため，以下のような別の方式を構成する必要があることに注意．）最初に，各参加者$i$の秘密情報$x_i$をSSを用いて，$n$人に分割する．次に，前と同様に，関数$f$を回路に対応させ，各ゲートごとの演算を秘密を分散させたまま行う．最後に，全員の計算結果を公開し，SSの機能より正しい計算結果を復元する．さて，先に述べたように，任意の回路の計算ができることを示すためには，AND及びNOTに対しそのような計算

ができることを示せば良い．ここで，AND 及び NOT は，mod 2 のかけ算及び +1 に対応するため，SS で“かけ算”及び“足し算”が秘密を分散させたままでできることを示せば良い．足し算については，**4.5** に書かれている方法でできることは明らかである．かけ算は，より複雑な手順が必要であるが，やはり実現可能である．

（2） 1/3 以内の不正者を許す方式

次に，$t$ 人の不正者がいても安全であるようなプロトコルを構成する．上記の基本プロトコルでは，SS を用いたが，ここでは，それを VSS に変更する．さらに，ここでは，前の方式と違って，公開鍵暗号系を前提としていないため，不正者を陽に特定できない．そこで，**4.6** で示したように，VSS を $t$ 重誤り訂正 BCH 符号と組み合わせることにより，$t<n/3$ のとき，$t$-安全であるようなマルチパーティプロトコルを構成できる．

## 6. あとがき

マルチパーティプロトコルがゼロ知識証明及びその応用である VSS や Oblivious Transfer を本質的な部分で用いることにより実現できることを示した．最初に，マルチパーティプロトコルの代表的な応用である選挙（無記名投票）を例に取りあげ，その具体的実現方法を示した．その後，一般的なマルチパーティプロトコルの研究動向とその実現例を概説した．

マルチパーティプロトコルは，ゼロ知識証明と並んで，暗号研究の大きなテーマの一つであり，また，ゼロ知識証明や VSS をはじめとして，これまで暗号の世界で開発された主な技法が多用されるという意味で，現代暗号研究の粋とも言えるものである．

残念ながら，現在，日本でこの分野の研究は，決して活発とは言えない．この記事により，この分野の研究を志そうとする方がおられるならば，著者らにとり望外の幸せである．

**謝辞** 有益なご意見をいただきました桜井幸一氏ならびに査読者に感謝いたします．

## 参考文献

1) Blum: Coin Flipping by Telephone, IEEE, COOMPCON, pp. 133-137 (1982).
2) Goldwasser, Micali and Rackoff: The Knowledge Complexity of Interactive Proof Systems, STOC '85, pp. 291-304 (1985), SIAM J. on Comp. Vol. 18, No. 1, pp. 186-208 (1989).
3) Cohen and Fischer: A Robust and Verifiable Cryptographically Secure Election Scheme, FOCS, pp. 372-382 (1985).
4) Benaloh and Yung: Distributing the Power of a Government to Enhance the Privacy of Voters, PODC, pp. 52-62 (1986).
5) Crépeau: A Zero Knowledge Poker Protocol That Achieves Confidentiality of Players' Strategy, or How to Achieve an Electronic Poker Face, CRYPTO '86, pp. 239-247 (1986).
6) Fiat and Shamir: How to Prove Yourself: Practical Splutions to Identification and Signature Problems, CRYPTO '86, pp. 186-194 (1986).
7) Goldreich, Micali and Wigderson: How to Play Any Mental Game, STOC '87, pp. 218-229 (1987).
8) Goldreich and Vainish: How to Solve Any Protocol Problem—an Efficiency Improvement, CRYPTO '87, pp. 73-86 (1987).
9) Chaum, Damgard and Graaf: Multiparty Computations Ensuring Privacy of Each Party's Input and Correctness of the Result, CRYPTO '87, pp. 87-119 (1987).
10) Galil, Harber and Yung: Cryptographic Computation: Secure Fault Tolerant Protocols, CRYPTO '87, pp. 135-155 (1987).
11) Ben-Or, Goldwasser and Wigderson: Completeness Theorems for Non-Cryptographic Fault Tolerant Distributed Computation, STOC '88, pp. 1-10 (1988).
12) Kurosawa, K., Katayama, Y., Ogata, W. and Tsujii, S.: General Public Key Residue Cryptosystems and Mental Poker Protocols, Eurocrypt '90 (1990).
13) Kurosawa, K. and Tsujii, S.: A General Method to Construct Public Key Residue Cryptosystems, Trans. IEICE, E 73, No. 7, pp. 1068-1072 (1990).
14) Blum: How to Prove a Theorem So No One Else Can Claim It, Proc. International Congress of Mathematics, pp. 1444-1451 (1986).
15) Benaloh: Cryptographic Capsules: A disjunctive Primitive for Interactive Protocols, CRYPTO '86, pp. 213-222 (1986).
16) Brassard and Crepeau: Zero Knowledge Simulation of Boolean Circuits, CRYPTO '86, pp. 223-233 (1986).
17) Chaum: Demonstrating That a Public Predicate Can Be Satisfied without Revealing Any Information about How, CRYPTO '86, pp. 195-199 (1986).
18) Brassard, Chaum and Crepeau: Minimum Disclosure Proofs of Knowledge, JCSS, pp. 156-166 (1988).
19) Goldreich, Micali and Wigderson: How to Prove All NP-statements in Zero Knowledge, and a Methodology of Cryptographic Protocol

Design, Crypt '86, pp. 171-185 (1986).
20) Brassard and Crepeau: Non-Transitive Transfer of Confidence: A Perfect Zero Knowledge Interactive Protocol for SAT and Beyond, FOCS, pp. 188-195 (1986).
21) Boyar and Peralta: On the Concrete Complexity of Zero-Knowldge Proofs, CRYPTO '89, pp. 507-525 (1989).
22) 黒沢, 辻井: Zero Knowledge Interactive Proof System for Modulo Operations, ISEC 90-40 (1990).
23) Benaloh: Secret Sharing Homomorphisms: Keeping Shares of a Secret, CRYPTO '86, pp. 251-260 (1986).
24) McEliece and Sarwarte: On Sharing Secrets and Reed Solomon Codes, Comm. ACM. Vol. 24, No. 9, pp. 583-584 (1981).
25) Shamir, A., Rivest, R. and Adleman, L.: Mental Porker, The Mathematical Gardner, Belmont, Cali, Wadsworth International, pp. 37-43 (1981).
26) Yao, A.: Protocols Secure Computation, Proc. of the 23rd FOCS, pp. 160-164 (1982).
27) Chor, B., Goldwasser, S., Micali, S. and Awerbuch, B.: Verifiable Secret Sharing and Achieving Simultaneity in the Presence of Faults, Proc. of the 26th FOCS, pp. 383-395 (1985).
28) Chaum, D., Crépeau, C. and Damgård.: Multiparty Unconditionally Secure Protocol, Proc. of the 20th STOC, pp. 11-19 (1988).
29) Rabin, T. and Ben-Or, M.: Verifiable Secret Sharing and Multiparty Protocols with Honest Majority, Proc. of the 21st STOC, pp. 73-85 (1989).
30) Beaver, D. and Goldwasser, D.: Multiparty Computations with Faulty Majorlty, Proc. of the 30th FOCS, pp. 468-473 (1989).
31) Beaver, D., Micali, S. and Rogaway, P.: The Round Complexity of Secure Protocols, Proc. of the 22nd STOC, pp. 503-513 (1990).
32) Beaver, D., Feigenbaum, J., Kilian, J. and Rogaway, P.: Security with Low Communication Overhead, Crypto '90 (1990).
33) Even, S., Goldreich, O. and Lempel, A.: A Randomized Protocol for Signing Contracts, Proc. of Crypto '82, pp. 205-210 (1982).
34) Yao, A. C.: How to Generate and Exchange Secrets, Proc. of 27th FOCS, pp. 162-167 (1986).

(平成3年3月22日受付)

**黒沢 馨**

昭和52年東京工業大学工学部電子工学科卒業. 昭和57年同大学院博士課程修了. 同年同大助手. 昭和60年同講師. 平成元年同助教授. 多種フロー問題, デジタル信号処理, 通信プロトコル, 情報セキュリティに関する研究に従事. 昭和55年度電子通信学会論文賞, 昭和60年度同学会篠原記念学術奨励賞各受賞. 電子情報通信学会, IEEE 学会各会員.

**岡本 龍明** (正会員)

1952年生. 1976年東京大学工学部計数工学科卒業. 1978年同大学院計数工学専攻修士課程修了. 同年NTT入社. 現在, 情報通信処理研究所主幹研究員, 入社以来, 主にネットワークアーキテクチャの研究, 自然言語処理の研究, 情報セキュリティの研究に従事. 工学博士. 電子情報通信学会会員.