数理解析研究所講究録
第1849巻 2013年 120-126

時間ドロボー問題の物質的ゼロ知識証明
(Physical Zero-Knowledge Proof Systems for Instant Insanity)

上田 圭祐 (Keisuke Ueda)* 西村治道 (Harumichi Nishimura)†

概要. 本論文は，ゼロ知識証明の物質的プロトコルを，どこまで簡単なものを用いて簡単に実装できるかについて研究する．Gradwohlらは数独問題に対するトランプを用いたプロトコルを考案し，そのプロトコルを実装するためには入力サイズに対して3倍のカードを必要としているが，本論文では時間ドロボー問題に対するトランプを用いたプロトコルで，入力サイズに対して1倍のカードで実装できるプロトコルを考案した．

1 序論

NP問題は，しばしば「答えがYesであることをチェックすることが容易なYes/No問題」であると説明される．このことは，以下の2人の登場人物をもとにしたシステムとして説明されることも多い．多項式時間の計算能力である検証者 (Verifier) と，計算能力に制限のない証明者 (Prover) と呼ばれる者がいるとする．検証者があるNP問題を解こうとしたときに，(自力で解くには難しそうであるが) 証明者の助けを借りることができるとする．問題の答えがYesの場合，証明者は証明を送り，検証者はその証明は正しいのかどうかを効率的に検証できる．一方で，答えがNoならそもそも証明は存在しない．よって，検証者は証明者が何を送ってこようとも証明が正しくないことをこれまた効率的に検証できるのである．

NPのシステムを拡張したものとして対話型証明がある．NP問題では証明者が検証者に証明を一方的に送るといったことをしていたが，対話型証明では検証者が証明者に質問をし，証明者はその答えを返すといったように，両者に双方向の対話を許す．またNP問題では検証者がYes/Noを100%正しく検証していたのに対し，対話型証明では検証者は高確率で検証できればよい．この対話型証明によって定義される計算量クラスとしてIP（Interactive Proofの略）がある．

対話型証明をさらに拡張したものとしてゼロ知識証明 (Zero Knowledge Proof) というものがある．対話型証明では，問題がYesかNoかを判定するためのヒントとなる情報を送ることが許されているが，ゼロ知識証明では，その情報を送らずに納得させなければならない．つまり，問題がYesであること以外の何の知識も伝えることのないような対話のみが許されるという条件をみたしたうえで，検証者がYes/Noを高確率で検証できる必要がある．

ゼロ知識証明のプロトコルには，一般的なプロトコルである暗号的プロトコルがある．これは2つのコンピュータがメッセージを交換して行うプロトコルである．しかしコンピュータ上で行うプロトコルは中身が見えず，素人から見れば分かりにくいものである．そういった目に見えない“コンピュータ上”のものよりも理解しやすいように，身近なものを使い，自分自身が参加することによって証明を理解し，納得することができる物質的プロトコルというものが考案されている（例えば文献 [1] およびその引用を参照）．これは素人にゼロ知識証明の概念を教え

*大阪府立大学大学院理学系研究科 (Graduate School of Science, Osaka Prefecture University)
†名古屋大学大学院情報科学研究科 (Graduate School of Information Science, Nagoya University)

るのに良い方法と考えられ，例えば [1] の物質的プロトコルではスクラッチカードやトランプなどを用いている．

このゼロ知識証明の先行研究として，NP 完全問題の 1 つである数独問題に対するゼロ知識証明のプロトコルがある [1]．しかし，文献 [1] で実際に与えられている物質的プロトコルは，ゼロ知識証明の概念のデモンストレーションとしては使用するカードの量などに課題があると考えられる．そこで，本論文では同じく NP 完全問題である時間ドロボー問題に対する物質的プロトコルを研究した．そして時間ドロボー問題に対する物質的プロトコルの結果が，先行研究よりも優れた点を含む結果となった．具体的には，数独問題に対するトランプカードを用いた物質的プロトコルを実装するためには，入力サイズに対して 3 倍のカードを必要としているのに対し，時間ドロボー問題に対しては入力サイズの 1 倍のカードで実装できるプロトコルを考案した．これを標準のサイズで考えると，標準の数独のサイズ (9×9) に対して 7 デッキのカードが必要であるが，時間ドロボー問題に対しては，標準の時間ドロボーのサイズ (ブロック 5 個) に対して 1 デッキで実装できるプロトコルを考案した．この結果は，「身近な問題を，身近なものを用いて，なるべく簡単にゼロ知識証明を教えたい」という物質的プロトコルの観点から見て，より望ましいものと考えられる．

2 ゼロ知識証明

先行研究の数独に対するプロトコルと，今回考案した時間ドロボーに対するプロトコルでは，「物質的ゼロ知識証明」という種類のゼロ知識証明が扱われている．この概念を説明するために，まずゼロ知識証明が満たすべき条件を記述し，文献 [4, 5, 6] を参考に完全及び計算量的ゼロ知識証明を説明する．その後，物質的ゼロ知識証明について述べる．

最初にプロトコルの完全性と健全性の概念を導入するため NP の定義を与える．

定義 1 （NP問題） L が NP に属するとは，以下の 2 つの条件をみたすある多項式時間アルゴリズム V および多項式 $p(n)$ が存在することをいう．
（完全性）$x \in L$ のとき，ある $y \in \{0,1\}^{p(n)}$ が存在して，$V(x,y) = \text{Yes}$ が成り立つ．
（健全性）$x \notin L$ のとき，すべての $y \in \{0,1\}^{p(n)}$ に対して，$V(x,y) = \text{No}$ が成り立つ．

NP 問題には，登場人物として計算能力が低い者 (検証者 Verifier) と，計算能力が高い者 (証明者 Prover) の 2 人がいる．検証者が，ある Yes/No 問題を解こうとしたとき，検証者は自力で解くのが難しかったとする．そこで，証明者がその問題の証明を教えてくれる．問題が Yes の場合に，Yes となるような証明が与えられて，その証明は正しいのかどうかを検証者が検証できるような問題を NP 問題という．

次に，この NP を拡張したものである対話型証明 (Interactive Proof) のクラス IP の説明をする．先ほどの NP 問題では証明者が証明を一方的に送るといったことをしたが，対話型証明では検証者が証明者に質問をし，その質問に対する答えを教えてもらうといったように，両者に双方向の対話を許す．また，NP 問題では検証者が Yes か No かを 100 %検証していたが，対話型証明は検証者が高確率で検証できればよいといったものである．

計算量的ゼロ知識証明には，対話型証明の完全性，健全性の条件に加え，ゼロ知識性という条件が必要である．ゼロ知識性について述べると，先ほどの NP 問題では，問題が Yes のときに証明者は検証者を納得させるために証明を送った．しかしこれは，Yes と判断するための情報を送ることが許されていた．ゼロ知識証明では，その情報を送らずに納得させなければならない．つまり，問題が Yes であること以外の何の知識も伝えることなく証明できるようなやりとりをしなければならない．このような手法をゼロ知識証明と言う．実際に情報が漏れているかどうかを調べるには，対話型証明における証明者と検証者の対話をシミュレートする多項式時間乱択アルゴリズム M(シミュレータ) を用いることで調べることが

できる．対話の内容を多項式時間でシミュレートできれば，証明者の情報は漏れていないと考えてよい．なぜなら，そのような対話で得られた情報は多項式時間乱択アルゴリズムである検証者自らがシミュレートできるような内容であるからである．このように対話型証明の内容をシミュレートできるとき，その対話型証明はゼロ知識であるという．これをふまえて，まず完全ゼロ知識証明について定義し [6]，そして計算量的ゼロ知識証明について説明する．

定義 2（完全ゼロ知識証明） 与えられた決定問題 Π に対する多項式時間対話型証明システムをもっていると仮定する．V^* を（だます可能性のある）検証者が質問を作るときに使う任意の多項式時間乱択アルゴリズムとする．（つまり，V^* で正直な検証者も不正をする検証者も表す．）証明者と V^* が Π の YES 入力 x について対話型証明を実行したときに生成されるすべての系列の集合を $\tau(V^*, x)$ と表記する．すべての V^* に対して，偽造系列を作り出す平均多項式時間乱択アルゴリズム $M^* = M^*(V^*)$（シミュレータ）が存在すると仮定する．偽造系列の集合を $F(V^*, x)$ と表記する．任意の系列 $T \in \tau(V^*, x)$ に対して，$p_\tau(T)$ を，T が対話型証明に参加した V^* によって作り出される系列である確率を表すものとする．$T \in F(V^*, x)$ に対しても同様に，$p_F(T)$ を，T が M^* によって作り出される（偽造の）系列である確率を表すものとする．$\tau(V^*, x) = F(V^*, x)$ が成立し，任意の $T \in \tau(V^*, x)$ に対して $p_{F,V^*}(T) = p_{\tau,V^*}(T)$ もまた成立するとき，対話型証明システムは（制限のない）**完全ゼロ知識**と呼ばれる．

つまり，検証者がプロトコルに参加したときに生成される系列と全く同じ確率分布で系列を作り出すシミュレータが存在するならば，対話型証明システムは完全ゼロ知識証明であるという．計算量的ゼロ知識証明では，系列の確率分布が全く同じである必要はなく，多項式時間アルゴリズムによって識別不可能であることだけが必要とされる．これ以外は，完全ゼロ知識証明として同様に定義される．ゼロ知識証明は，IP の定義の完全性，健全性，それとゼロ知識性の 3 条件を満たす必要がある．

物質的プロトコルのゼロ知識性については，文献 [1]（あるいはその引用文献 [3]）にならって，証明者役とシミュレータ役の見分けがつかない (系列とその確率分布が同じ) ことに加え，検証者がプロトコルを逸脱したときは知識が漏れる代わりに検証者が不正をしたことが発覚するゼロ知識であると定義する．（厳密にはこのようなゼロ知識性は暗号的プロトコルのゼロ知識性（多項式時間検証者がプロトコルを逸脱しても知識は漏れない）より弱いものである．）

3 数独に対するプロトコル

Gradwohl らによる先行研究である数独問題[1]に対するトランプを用いた物質的プロトコル [1] を紹介する．なお，検証者を V，証明者を P として記述する．問題の定義は文献 [2] を参考にした．

問題 SDK(数独)
入力： $n \times n$ のグリッドが $k \times k$ のサブグリッド $(n = k^2)$ に分割されていて，いくつかのセルには数字が埋められている．
出力： 空いたセルに $1, \cdots, n$ の数字を埋める．このとき各行・各列・各サブグリッドが $1, \cdots, n$ の数字をすべて含むようにできるか？

数独に対するトランプを用いた物質的プロトコルは以下のようである．

SDK1 (Gradwohl ら [1])

- P はそれぞれのセルに 3 枚のカードを置く．すでに埋まっているセルには，その値に一致する 3 枚のカードを，表にした状態で置く．
- V は行/列/サブグリッドそれぞれにおいて，それぞれのセルからランダムに 1 枚を選ぶ．

[1]数独はニコリ社の商標登録である．

- P は行/列/サブグリッドそれぞれに対して，リクエストされたカードが入っているパケットを作る．そのとき，集められた $3n$ 個のパケットをそれぞれ別々にシャッフルし，シャッフルされたパケットを V へ渡す．
- V は各パケットのカードすべてを表にし，各パケットがすべての数字を 1 つずつ含んでいるか確かめる．含んでいるなら受理．

数独のトランプを用いた結果をまとめると次の表 1 のようになる．プロトコル SDK1 では，カードをシャッフルする回数が $3n$ 回ある．これを減らす目的で文献 [1] ではプロトコル SDK2 も与えられている．なお，c は 2 以上の整数である．プロトコル SDK2 の説明，それぞれのプロトコルの解析は省略する．

	カード数	シャッフル数	健全性誤り
SDK1	$3n^2$	$3n$	$\frac{1}{9}$
SDK2	$3n^2$	$c-1$	$\frac{1}{9}+\frac{8}{9c}$

表 1: 文献 [1] の数独に対するプロトコル

4 時間ドロボーに対するプロトコルと結果

本章では，時間ドロボー問題に対するトランプを用いた物質的プロトコルを提案し，文献 [1] の結果と比較する．なお，検証者を V，証明者を P として記述する．問題の定義は文献 [2] を参考にした．

問題 INS(時間ドロボー)
入力：n 個の立方体．ただし各面は n 色の中の 1 色で塗られている．
出力：すべての立方体を一列に積み重ねる．このとき積み重ねて現れる 4 つの面のそれぞれに，各色がちょうど 1 回ずつ現れるようにできるか？

提案プロトコル INS1 は以下のようである．

INS1 (INS に対するトランプ使用プロトコル)

- P は立方体に n 番まで番号を付け，この情報を V と共有する．
- P はカードを展開図に対応するように 6 枚セットの形に裏にして n セット置く．またそのカードを置く前に各セットに対して，側面に対応する部分として順番に 4 つのカードを指示する可能性があるので，指示する順番をランダムに決めておく．
- V は次の 2 つからランダムに 1 つ行う．
  (a) V はカードをすべて表にする．この配置とそれぞれの立方体とを順番ごとに比べ，矛盾があれば拒否する．
  (b) P に各セットに対して，側面に対応する順番通りに 4 つのカードを指示してもらい，V は指示された通りにカードを集め，順番通りに横に並べる．n 番目までこの動作を繰り返した後，縦にカードを集めて 4 つのパケットを作る．これを P へ渡し，P はそのパケットをシャッフルして V へ返す．V はそれを受け取り，表にし，各パケットに各色がちょうど 1 回ずつ現れているかチェックする．現れていないなら拒否する．また，指示されたカードの配置が側面に矛盾してる場合も拒否する．

このプロトコルの解析は次の通りである．

(完全性の証明) 入力の答えが Yes のとき，各色がちょうど 1 回ずつ現れるように並べることができる．その解を平面で考えたとき，1 つの解に対して時計回りに 4 種類，(天地を替えることによって) 反時計周りに 4 種類の合計 8 種類の解がある．よって解は最低 8 種類ある．P はまず最初に，最低 8 種類の解

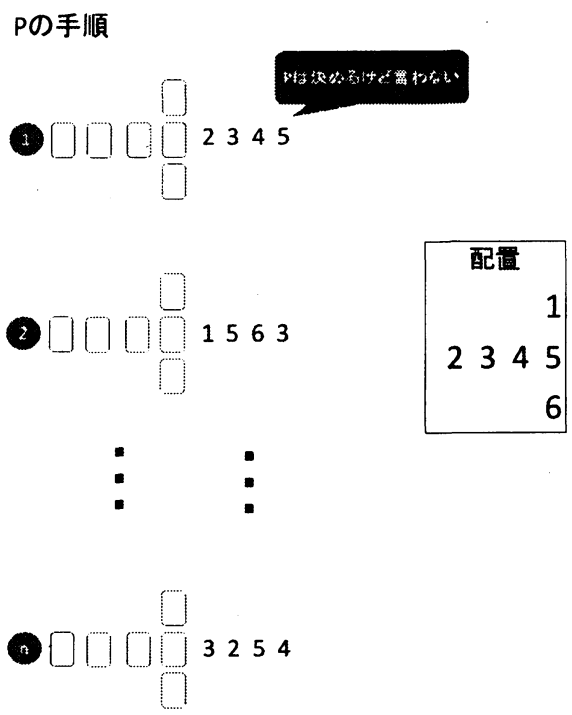

図 1: プロトコル INS1 の証明者の手順

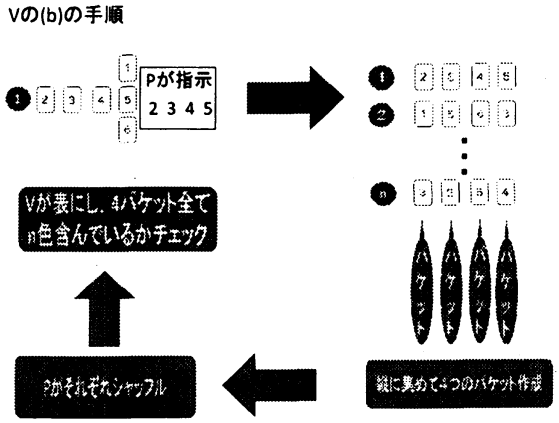

図 2: プロトコル INS1 の検証者が (b) を選んだときの手順

の中から 1 つランダムに選び，解を定める．次に，指示する順番をランダムに定める．その順番と解に沿ってカードを置いていく．まだカードを置いてない部分は，入力に矛盾がないようにすれば一意に決まるので，その通りに置いていく．このように置くと，V が手順 (a),(b) どちらを選んでも受理する．よって V は確率 1 で受理する．

定理 1 （健全性） このプロトコルにおいて，各色がちょうど 1 回ずつ現れるように並べることができないとき，$\Pr[V_{accept}] \leq \frac{1}{2}$ である．

(健全性の証明) 入力の答えが No のとき，(a) と (b) 両方受理させる方法がないことを示す．

- **(a) を確実に受理させるようにするとき** (a) を確実に受理させるためには，入力に矛盾のない配置に置けばよい．しかしこの場合，立方体に対応するカードの配置として 1 つも嘘をつくことができない (嘘の立方体を混ぜることができない)．よって健全性の場合，積み重ねて現れる 4 つの面すべてを，どんな側面の組み合わせを選んでも各色がちょうど 1 回ずつ現れるようにすることができない．したがって，この場合は (b) で拒否される．

- **(b) を確実に受理させるようにするとき** (b) を確実に受理させるためには，4 つの面すべてに各色がちょうど 1 回ずつ現れるようにし，かつ正しい側面の組み合わせを選べばよい．入力が No のとき，これを満たすためには立方体に対応するカード配置で嘘をつかなければならない (嘘の立方体を混ぜなければならない)．よって入力に矛盾する立方体を使用する．したがって，この場合は (a) で拒否される．

(ゼロ知識性の証明) シミュレータ M は，V が (a), (b) どちらを選ぶか予測する．

- **(a) を選ぶと予測するとき** M は，まず立方体に n 番まで番号を付け，この情報を V と共有す

る．その後，入力に矛盾のないように展開図の形にして n セット置く．置き方は，入力に矛盾のない範囲でランダムに置く．4つのカードを指示する順番もランダムに決める．P と M が置く配置の分布を比較する．1セットで比較する．P がとる手順を見ると，側面1組が決まった後，可能な側面の組み合わせ24パターン (最初6か所のうち1つ決め，次に4か所から1つ決めたら側面が決まるので，$6\times 4=24$) のうちからランダムに1つ選び，そこでカードの配置が決まる．M がとる手順を見ると，ある1つの配置に着目して，最初に立方体6面のうちから1つをランダムに選び，次にその隣の配置に対して立方体4面のうちから1つをランダムに選ぶと，残りの配置が決まる．よって $6\times 4=24$ パターンのうちからランダムに1つ選べば配置が決まる．よって，P と M が置く配置の分布はどちらも入力に矛盾のない24パターンの中からランダムに選ばれるので，系列と確率分布は等しい．

- **(b) を選ぶと予測するとき** M は，まず立方体に n 番まで番号を付け，この情報を V と共有する．その後，V へ指示するカードを n セットまで，側面に矛盾のない範囲でランダムに決める．ランダムに決めた割り当てに対して，集められた4つのパケットそれぞれが，各色がちょうど1回ずつ現れるようにカードを置く．置き方は，各色がちょうど1回ずつ現れる範囲でランダムに置く．V が (b) を選ぶとき，M は前もって決めていた指示の順番通りに指示する．P と M によるパケットにカードを集められた後，集められてないカードの配置の分布を比較する．1セットで比較する．P がとる手順を見ると，解を決めた後，可能な側面の組み合わせ24パターンのうちからランダムに1つ選ぶ (これは，P が V を納得させたいため，シミュレータと区別がつかないようにランダムに選ぶ手段をとる)．M がとる手順を見ると，可能な側面の組み合わせ24パターンのうちからランダムに1つ選び，そこから解を決め，カードを置く．すなわち，集められてないカードの配置は同じ分布になり，また最後に開示するパケットの分布も等しい．よって，P と M による系列の確率分布は等しい．

  M の予測が外れたとき，最初からやり直す．また，V がコインにインチキをした場合を考える．このとき，V が (a) を選ぶ確率と (b) を選ぶ確率が異なってくるが，シミュレータ M は予想が外れると最初からやり直し，予想が当たると上記のような動作を行い，この動作からは P と M の見分けがつかない．よってサイコロにインチキをしても P と M の見分けがつくことはない．

数独のプロトコルと今回のプロトコルを比べると，次の表2のようになる．文献 [1] の数独プロトコルと比べ，健全性誤り確率では劣るが，SDK と INS の入力サイズがそれぞれ n^2，$6n$ であるので，入力サイズに対して必要なカード枚数を3倍から1倍に減らせている．なお，c は2以上の整数である．

	カード数	シャッフル数	健全性誤り
SDK1	$3n^2$	$3n$	$\frac{1}{9}$
SDK2	$3n^2$	$c-1$	$\frac{1}{9}+\frac{8}{9c}$
INS1	$6n$	4	$\frac{1}{2}$

表 2: 数独 [1] および時間ドロボーのプロトコル

次に，実装することを想定して，標準的な値でプロトコルを比較する．時間ドロボーは，$n=5$ のものがおもちゃとしてハナヤマから販売されている．数独は主に入力 9×9 で売られていることが多いので，それぞれ標準的な値としてこの数値で比較する．このとき，次のような定理が得られた．ここでいう「扱いやすさ」とは，トランプと色の対応が実演した場合に何度も対応を確認しなくてよいような自然な対応になっているかを指している．

定理 2 $n=5$ のとき，このプロトコルはトランプ1

デッキで，かつトランプを扱いやすい状態で実装できる．

(証明) $n=5$ のとき，5種類の数字さえ扱えれば色を表現できる．5種類の数字を，トランプ1デッキで $1,\ldots,5$ の各4枚，$6,\ldots,10$ は，5で割った余りの数 (mod 5) として考え，$1,\ldots,5$ の代わりとする．このようにすると，トランプ1デッキで $1,\ldots,5$ それぞれ8枚用意することができる．これで色を表現していく．

しかし，色が9つ以上現れるケースは，これでカバーできない．ここで，各色は少なくとも4つ以上現れるという事実を考える (3つ以下の色があると，各色がちょうど1回ずつ現れるように並べることができないことがすぐわかるため，問題にならない)．これをもとに考えると，$n=5$ のとき，1種類または2種類の色が9つ以上現れることはあるが，3種類以上の色が同時に9つ以上現れることはない．

9つ以上現れる色が1種類のとき，余っている11，12，13のカードを足りない部分として補う (例えば，11以上の値のカードは水色とする等)．9つ以上現れる色が2種類のとき，この場合は3種類の色が4つ現れ，2種類の色が9つ現れるパターンしかない．11，12，13のカードを足りない部分として補う (例えば，11を水色，12を赤色とする等)．

これらの方法は余りカードの11，12，13の枚数で十分行える．よって $n=5$ のとき，このプロトコルはトランプ1デッキで，かつトランプを扱いやすい状態で実装できる．

数独の入力が 9×9，時間ドロボーの入力 (ブロック数) が5個の場合で比べると，次のようになる．

	デッキ数	シャッフル数	健全性誤り
SDK1	7	27	$\frac{1}{9}$
SDK2	7	$c-1$	$\frac{1}{9}+\frac{8}{9c}$
INS1	1	4	$\frac{1}{2}$

表 3: 標準な数値でのプロトコルの比較

なお，時間ドロボーを9ブロックとした場合は，デッキ数2，シャッフル数4，健全性誤り $\frac{1}{2}$ となる．健全性誤り確率の改善，そして他の問題に関してのプロトコルの考案などが今後の課題である．

参考文献

[1] R. Gradwohl, M. Naor, B. Pinkas and G. N. Rothblum : Cryptographic and physical zero-knowledge proof systems for solutions of Sudoku puzzles. Theory Comput. Syst. 44(2): 245–268 (2009).

[2] ロバート·A·ハーン, エリック·D·ドメイン (著), 上原 隆平 (訳), ゲームとパズルの計算量, 近代科学社, 2011. (原著 R. A. Hearn and E. D. Demain: Games, Puzzles, and Computation, A K Peters/CRC Press, 2009.)

[3] T. Moran and M. Naor : Basing cryptographic protocols on tamper-evident seals. In: Proceedings of the 32nd International Colloquium on Automata, Languages and Programming (ICALP 2005). Lecture Notes in Computer Science, vol. 3580, pp. 285–297, Springer, 2005.

[4] 岡本 龍明, 山本 博資 (著) , 現代暗号 (シリーズ・情報科学の数学), 産業図書, 1997.

[5] マイケル・シプサ (著), 太田 和夫, 田中 圭介, 阿部 正幸, 植田 広樹, 藤岡 淳, 渡辺 治 (訳) , 計算理論の基礎 [原著第2版], 共立出版, 2008. (原著 M. Sipser: Introduction to the Theory of Computation, Course Technology Ptr (Sd), 1996.)

[6] ダグラス・R・スティンソン (著), 桜井 幸一, 古屋 聡一, 檀浦 詠介, 山家 明男, 赤星 信博, 佐野 文彦, 山根 義則 (訳) , 暗号理論の基礎, 共立出版, 1996. (原著 D. R. Stinson: Cryptography: Theory and Practice (Discrete Mathematics and Its Applications), CRC Press, 1995.)