一次不定方程式に基づくゼロ知識対話証明

八木沢 正博†

「ゼロ知識対話証明」の具体的な実現法は，大きな素数の積を法とする平方剰余問題や離散的対数問題に基づくものや，二つのグラフの同形問題に基づくもの等多くの事例が発表されている．本論文でディオファンタスの一次不定方程式に基づく一方向関数を利用したゼロ知識対話証明の具体的な実現方法を提案する．本方法では，扱うビット長，演算量が小さくなるためゼロ知識対話証明やマルチパーティプロトコルを実現するのに必要な計算量が，大きな素数の積を法とする多次剰余暗号系を利用した方法に比較して，かなり少なくなる利点がある．系の構成は，次のようになる．正整数 a_i を係数，C を与えられた整数，t_i を未知数とするディオファンタスの一次不定方程式 $C=a_1t_1+\cdots+a_nt_n$ を利用して，$0\leqq t_i<P$ (P は適当な正整数) なる t_i を入力，C を出力とする問題を構築する．ここで，$T=(t_1,\cdots,t_n)$，$C=f(T)$ とおく．乱数 $D=(d_1,\cdots,d_n)$ を選び，$z_1=t_1+d_1 \bmod P,\cdots,z_n=t_n+d_n \bmod P$ とする．このとき，$t_i+d_i-z_i=0$ or P $(i=1,\cdots,n)$ であり，$z_1,\cdots,z_n$ にはランダム自己帰着性がある．この z_i を用いてゼロ知識対話証明を実現できる．

Zero-Knowledge Interactive Proof Systems Based on Linear Indeterminate Equation

MASASHIRO YAGISAWA†

Zero-knowledge interactive proof systems based on linear indeterminate equation is proposed in this paper. The operations do not require so much computational complexity in the ZKIP systems as compared with residue cryptosystems. We can generate a public-key cryptosystem by using a linear indeterminate equation as a one-way function. The cryptosystem is constructed as follows. With $n+1$ prime numbers P_i $(i=0,\cdots,n)$, and n positive integers k_i $(i=1,\cdots,n)$, we get b_i such that $b_i=k_i\,P_0\cdot\cdot P_{i-1}$ $(i=1,\cdots,n)$. We find positive integers $a_i<b_i$ $(i=1,\cdots,n)$, R and β such that $R>b_1P_1+\cdots+b_nP_n$, $a_i\equiv b_i\beta \bmod R$ $(i=1,\cdots,n)$, where β is relatively prime to R, and $R>a_1P_1+\cdots+a_nP_n$. We construct a cryptosystem based on an indeterminate equation such that $C=a_1x_1+\cdots+a_nx_n$ where $0\leqq x_i<P_i$ $(i=1,\cdots,n)$. x_i are plaintexts and C is ciphertext in this cryptosystem.

1. はじめに

「ゼロ知識対話証明[1),2)]」の具体的な実現法は，大きな素数の積を法とする平方剰余問題や離散的対数問題に基づくものや，二つのグラフの同形問題に基づくもの等多くの事例が発表されている．本論文でディオファンタスの一次不定方程式に基づく一方向関数[3)]を利用したゼロ知識対話証明の具体的な実現方法を提案する．本方法では，扱うビット長，演算量が小さくなるため，ゼロ知識対話証明やマルチパーティプロトコルを実現するのに必要な計算量が，大きな素数の積を法とする多次剰余暗号系を利用した方法に比較して，かなり少なくなる利点がある．本論文の構成は，次のようになる．2章で，今回採用する一方向関数を説明し，具体的に構築する方法について述べる．3.1節では，ゼロ知識対話証明法の基本プロトコルを説明する．3.2節では基本プロトコルを使ってゼロ知識対話証明法を構築する．4章ではゼロ知識対話証明を用いたマルチパーティプロトコル[4)]の実現例をいくつか述べる．5章では，本ゼロ知識対話証明法の優位性について述べ，6章では，今後の課題について述べる．

† 昭和エンジニアリング(株)
Showa Engineering Corporation

2. 一方向関数の構築[5)]

$$P_0=1 \tag{2.1}$$

$$P_i\ (i=1,\cdots,n) \tag{2.2}$$

は大きさがほぼ等しい素数とする．つまり，

$$o(P_1)=o(P_2)=\cdots=o(P_n)=A,\ (A \text{ は整数}) \tag{2.3}$$

暗号化鍵を公開するユーザ u_A は正定数，b_i $(i=1,\cdots,n)$ を，次のように生成する．

$$b_1=k_1P_0 \tag{2.4a}$$
$$b_2=k_2P_0P_1 \tag{2.4b}$$
$$\cdots$$
$$\cdots$$
$$b_n=k_nP_0P_1\cdots P_{n-1} \tag{2.4c}$$

ここで，k_i $(i=1,\cdots,n)$ は

$$o(k_1)=A^{n-1} \tag{2.5a}$$
$$o(k_2)=A^{n-2} \tag{2.5b}$$
$$\cdots$$
$$\cdots$$
$$o(k_{n-1})=A \tag{2.5c}$$
$$k_n=1 \tag{2.5d}$$

のように選ぶ．

したがって，b_1 の大きさは

$$o(b_1)=o(b_2)=\cdots=o(b_n)=A^{n-1} \tag{2.6}$$

となる．

平文 t_i $(i=1,\cdots,n)$ の定義域を次のように定める．α は正整数．

$$0\leqq t_1<P \tag{2.7}$$
$$P=\min_{0<i<n} P_i-\alpha \tag{2.8}$$
$$b_1P_1+\cdots+b_nP_n<R \tag{2.9}$$

なる整数 R を選び，R と互いに素な正整数 β を次のように選ぶ．

(2.4a)～(2.4c) 式の b_i を β, R でモード変換したものを a_i とする．

$$a_i\equiv\beta b_i \mod R \tag{2.10}$$

このとき，

$$\sum_{i=1}^{n} a_iP<R \tag{2.11}$$

となるように，LLL のアルゴリズム[6]を用いて，β, R を選ぶ．

つまり，次の基底ベクトルをもつ $n+1$ 次元の整数ラティスを構成する．

$$v_1=(R,0,\cdots,0,0) \tag{2.12a}$$
$$v_2=(0,R,\cdots,0,0) \tag{2.12b}$$
$$\cdots$$
$$v_n=(0,0,\cdots,R,0) \tag{2.12c}$$
$$v_{n+1}=(b_1,b_2,\cdots,b_n,1) \tag{2.12d}$$

次に，このラティスで最も短いベクトル v を見つける．

$$v=(\beta b_1-L_1R,\ \beta b_2-L_2R,\cdots,\ \beta b_n-L_nR,\ \beta) \tag{2.13}$$

L_i $(i=1,\cdots,n)$ は整数である．

v を見つけるのに必要な演算時間は，

$$o(n^6(\log R)^3) \tag{2.14}$$

である．実際には，最短なベクトルを求める必要はなく，(2.11) 式を満たす a_i が求まれば十分である．また，R の値は (2.9) 式を満たす素数であれば任意に選べるので，β, R の候補は多数存在すると思われる．P_i は秘密にするため，t_i の定義域を明確に公開することはできないが，P を公開する．

パラメータのサイズとして，次のサイズを推奨する．

$$n=5 \tag{2.15}$$
$$o(A)=10^{30} \tag{2.16}$$
$$o(R)=5\cdot10^{150} \tag{2.17}$$
$$o(a_i)=o(b_i)=10^{120}\ (i=1,\cdots,n) \tag{2.18}$$
$$o(P)=10^{30} \tag{2.19}$$
$$o(\alpha)=10^{25} \tag{2.20}$$

公開するパラメータ K_E は，次のようである．

$$K_E=[a_i\ (i=1,\cdots,n),\ P] \tag{2.21}$$

[暗号化] 公開されている a_i と平文 t_i から，

$$C=a_1t_1+\cdots+a_nt_n \tag{2.22}$$

を計算し，暗号文 C のみを送信する．

これに要する計算量は，5個の乗算と4個の加算のみである．K_E が必要とする容量は 2,260 bit となる．

ここで，$T(t_1,\cdots,t_n)$, $C=f(T)$ とおくと一方向関数 f が得られる．つまり，T から C を計算することはきわめて容易であるが，C から T を求めることは，計算量の上からみて非常に困難である．

[復号化] 正規の受信者が C を受信すると，$\beta^{-1}\bmod R$ を用いて C' を求める．

$$C'\equiv C\ \beta^{-1} \mod R \tag{2.23}$$
$$\equiv(a_1t_1+\cdots+a_nt_n)\beta^{-1} \mod R$$
$$\equiv b_1t_1+\cdots+b_nt_n \mod R$$
$$=b_1t_1+\cdots+b_nt_n \tag{2.24}$$

したがって，

$$C' \mod P_1\equiv b_1t_1\equiv k_1t_1 \mod P_1 \tag{2.25}$$
$$C'\ k_1^{-1} \mod P_1\equiv t_1 \tag{2.26}$$
$$C'-b_1t_1 \mod P_2\equiv k_2P_1t_2 \mod P_2 \tag{2.27}$$
$$(C'-b_1t_1)k_2^{-1}P_1^{-1} \mod P_2\equiv t_2 \tag{2.28}$$

以下同様にして，t_i $(i=1,\cdots,n-1)$ を求めることができる．

$$t_ib_i\equiv C'-b_1t_1-\cdots-b_{i-1}t_{i-1} \mod P_i \tag{2.29}$$

だから，

$$t_i\equiv(C'-b_1t_1-\cdots-b_{i-1}t_{i-1})k_1^{-1}P_1^{-1}\cdots P_{i-1}^{-1} \mod P_i \tag{2.30}$$

最後に，

$$t_n=(C'-b_1t_1-\cdots-b_{i-1}t_{i-1})/b_n \quad (2.31)$$

より，t_n を求める．復号化鍵 K_D は，

$$K_D=[\beta, R, b_i, P_i \ (i=1,\cdots,n)] \quad (2.32)$$

であるが，各 b_i を公開されている a_i の値から求めることも可能である．

(2.15)～(2.20) 式のパラメータを採用すると，復号化に必要な計算量は，R を法とするモジュラー乗算1個，P_i を法とするモジュラー乗算6個，4個の減算，1個の除算のみである．

K_D が必要とする容量は 3,595 bit である．

[以下余白]

3. ゼロ知識対話証明

3.1 基本プロトコル

2章で述べた一方向関数にはランダム自己帰着性がないため，次のような演算を導入する．

ここで，$T=(t_1,\cdots,t_n)$, $C=f(T)$ とおく．乱数 $D=(d_1,\cdots,d_n)$ を選び，

$$Z_1=t_1+d_1 \mod P, \quad (3.1)$$

$$\cdots$$

$$\cdots$$

$$z_n=t_n+d_n \mod P, \quad (3.2)$$

とする．このように，T と D から Z を生成する演算を

$$Z=T \sharp D \quad (3.3)$$

と表す．このとき $z_1, z_2, \cdots, z_n$ にはランダム自己帰着性がある．

次に示すプロトコルを基本プロトコルとする．

$C=f(T)$ つまり C を与える T を証明者 P が知っていることを検証者 V に示すゼロ知識証明のプロトコル P と V の間で次のやりとりを k 回繰り返す．

(step 1) P は乱数 D を選び $X=f(D)$ を計算し，X を V に送る．

(step 2) V は $e\in\{0,1\}$ を二者択一的にランダムに選び，e を P に送る．

(step 3) P は次の Y を V に送る．

$$Y=\begin{cases} D & \text{if } e=0 \\ Z=T\sharp D & \text{if } e=1 \end{cases} \quad (3.4)$$

(step 4) V は次式が成立するか検査する．

$$X=f(Y) \quad \text{if } e=0 \quad (3.5)$$

$$C+X=f(Y)+L \quad \text{if } e=1 \quad (3.6)$$

$$L=P(a_1e_1+\cdots+a_ne_n) \quad (3.7)$$

$$e_i\in\{0,1\} \quad (i=1,\cdots,n)$$

L の値は0から $a_1+\cdots+a_n$ の範囲の 2^n 通りであり，一覧表をあらかじめ作成しておき，上式を満足する L が存在するか否かチェックする．

3.2 ゼロ知識対話証明法

証明者 P が秘密鍵 $k_D=[\beta, R, b_i, P_i(i=1,\cdots,n)]$ を知っていることを検証者 V に示すゼロ知識対話証明のプロトコル（プロトコル1）

(step 1) V は T をランダムに選んで，$C=f(T)$ を計算し，C を P に送る．

(step 2) 基本プロトコルを用いて，C を与える T の値を V が知っていることを P に納得させる．

(step 3) P は秘密鍵 k_D を知っているので，C の値から T を計算する．

(step 4) 基本プロトコルを用いて，C を与える T の値を P が知っていることを V に納得させる．

4. 応 用 例

わかりやすいマルチパーティプロトコルの例として，選挙のプロトコルを示す．

4.1 センターが一つの方式

[1] センタは公開鍵 $K_E=[a_i(i=1,\cdots,n), P]$ を公開する．

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

$$z_i=a_1m_i+a_2x_2+\cdots+a_nx_n \quad (4.1)$$

として公開する．ただし，$x_2,\cdots,x_n$ は乱数であり，投票者数を h として，

$$h<P \quad (4.2)$$

[3] センタは，それらを復号し，

$$M=m_1+m_2+\cdots+m_h \quad (4.3)$$

を投票結果として公開する．

各段に対応する verification は次のとおりである．

[1] センタは秘密鍵 $k_D=[\beta, R, b_i, P_i(i=1,\cdots,n)]$ を知っていることをプロトコル1を用いて，ゼロ知識対話証明する．

[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証明で示す．

(step 1) $A(=i)$ は

$$z_i=a_im_i+a_2x_{2i}+\cdots+a_nX_{ni} \quad (4.4)$$

ただし，$x_{2i},\cdots,x_{ni}$ は乱数であり，

$0\leqq x_{2i},\cdots,x_{ni}<P/h$ h：投票者数

より，z_i を B に送る．

以下の step 2～5 を $j=1,\cdots,k$ について繰り返す．

(step 2) A は，乱数 $s_{j2},\cdot\cdot,s_{jn},t_{j2},\cdot\cdot,t_{jn}$ を選び

$$u_j=a_2s_{j2}+\cdots+a_ns_{jn} \quad (4.5)$$

$$v_j=a_1+a_2t_{j2}+\cdots+a_nt_{jn} \quad (4.6)$$

を計算し，この u_j, v_j をランダムな順番で B に送る．

(step 3) B は，ランダムに $e=0$ or 1 を A に送る．

(step 4) A は，

$e=0$ であれば，$s_{j2}, \cdots, s_{jn}$; $t_{j2}, \cdots, t_{jn}$ を B に送る．

$e=1$ であれば u_j, v_j のうち z_i と等価な（平文が同じ）ほうを w として，

$$z_i-w=a_2x'_2+\cdots+a_nx'_n \quad (4.7)$$

を満たす $x'_2, \cdots, x'_n$ を B に送る．

(step 5) B は送られてきたものから次のようにチェックする．

$e=0$ のとき，

$$a_2s_{j2}+\cdots+a_ns_{jn}=u_j \quad (4.8)$$

$$a_1+a_2t_{j2}+\cdots+a_nt_{jn}=v_j \quad (4.9)$$

が成立していることを確認する．

$e=1$ のとき，送られてきた u_j, v_j のいずれかを w として，

$$a_2x'_2+\cdots+a_nx'_n=z_i-w \quad (4.10)$$

が成立していることを確認する．

[3]

(step 1) A（センタ）は $z_1, \cdots, z_h$ から，m_i, x_{2i}, $\cdots$, $x_{ni}(i=1, \cdots, h)$ を復号化し，

$$x_1=0 \quad (4.11)$$

$$x_2=x_{21}+\cdots+x_{2h}<P \quad (4.12)$$

$$x_n=x_{n1}+\cdots+x_{nh}<P \quad (4.13)$$

を計算して

$$y=(M, x_2, \cdots, x_n) \quad (4.14)$$

なる y を B に送る．

(step 2) B は

$$z_1+\cdots+z_h=a_1M+a_2x_2+\cdots+a_nx_n \quad (4.15)$$

であることをチェックする．

4.2 センタが複数の方式

前章では，センタに各投票者の投票内容がわかってしまう．センタを複数にするとこの欠点を除去できる．センタ数を d, 投票者数を h とする．

[1] センタ j は公開鍵 $K_{Ej}=[a_{ij}(i=1, \cdots, n), P^j]$ を公開する．$(j=1, \cdots, d)$

[2] 投票者 i は自分の投票 m_i ($m_i=0$ or 1) を

$$m_i=m_{i1}+m_{i2}+\cdots+m_{id} \mod h', h'>h \quad (4.16)$$

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

$$z_{ij}=a_{1j}m_{ij}+a_{2j}x_{2ij}+\cdots+a_{nj}x_{nij} \quad (4.17)$$

として，公開する．ただし，$x_{2ij}, \cdots, x_{nij}$ 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，

$$M_j=m_{1j}+m_{2j}+\cdots+m_{hj} \mod h' \quad (4.18)$$

として，公開する．投票結果は，

$$M=M_1+M_2+\cdots+M_d \mod h' \quad (4.19)$$

で与えられる．

各 step の verification は省略する．

4.3 Verification secret sharing

[1] 分配者は，自分の秘密 m を定数項とするランダムな $(k-1)$ 次の多項式 $f(x)$ を選ぶ．

$$f(x)=m+g_1x+g_2x^2+\cdots+g_{k-1}x^{k-1} \mod q \quad (4.20)$$

分配者は，各分割保持者 i $(i=1, \cdots, N)$ に $f(i)$ を配る．（ここで，q は $m<q<P^*/2$ となるような適当な素数とする．P^* は，各分割保持者 i の公開鍵の P^i の最小値とする．）

[2] N 個に分割された分割情報 $f(i)$ $(i=1, \cdots, N)$ 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，不正な分配者は，乱数を $f(i)$ として配る可能性がある．このとき，k 人の分割保持者が復元して得られる結果と別の k 人の分割保持者が復元して得られる結果が異なる．これを防ぐには，各分配者が正しいプロトコルに従っていることをゼロ知識対話証明で示せばよい．（つまり，これが VSS である．）

i 番目の分割保持者の暗号系の公開鍵 $K_{Ei}=[a_{ji}(j=1, \cdots, n), P^i]$ とする．

(step 1) 分配者は，$s_i=f(i)$ を i 番目の分割保持者の暗号系で暗号化し，

$$z_i=a_{1i}s_i+a_{2i}x_{2i}+\cdots+a_{ni}x_{ni} \quad (4.21)$$

を公開する．ただし，x_{ji} は乱数．

i 番目の分割保持者は，これを復号し，自分の share s_i を得る．

以下は，

$$L=\{z_1, \cdots, z_N | z_i=a_{1i}s_i+a_{2i}x_{2i}+\cdots+a_{ni}x_{ni}, s_i=f(i)\} \quad (4.22)$$

に対するゼロ知識対話証明である．

以下の step 2〜4 を $[\log_2P^i]$ 回繰り返す．

(step 2) 分配者は，ランダムは $k-1$ 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 番目の分割保持者

の暗号系で暗号化し，

$$z'_i = a_{1i}s'_i + a_{2i}x'_{2i} + \cdots + a_{ni}x'_{ni} \qquad (4.23)$$

を公開する．ただし，$0 \leqq x'_{ji} < P^i/2$ は乱数．

(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，何らかの合意の上にランダムに $e=0$ or 1 を分配者に送る．)

(step 4)

$e=0$ のとき，分配者は，s'_i と x'_{ji} をすべてオープンにし，f' が高々 $k-1$ 次の多項式であることを示す．

$e=1$ のとき，分配者は

$$z_i + z'_i = a_{1i}(s_i + s'_i) + a_{2i}(x_{2i} + x'_{2i}) + \cdots + a_{ni}(x_{ni} + x'_{ni}) \qquad (4.24)$$

を満たす．

$$t_{1i} = s_i + s'_i \qquad (4.25)$$

$$\cdots$$

$$t_{ni} = x_{ni} + x'_{ni} \qquad (4.26)$$

をすべて $(i=1, \cdots, N)$ をオープンし，$f+f'$ が高々 $k-1$ 次の多項式であることを示す．

5. 一次不定方程式に基づく方法の優位性

2, 3, 4 章の実現例からもわかるように，本方法は多次剰余暗号系[4] に比較して，暗号化・復号化に要する計算量が少なくて済む．特に，VSS においては平文 s_i の値が大きくなるとその差が顕著になる．この事実は，次のように説明される．

[多次剰余暗号系]

秘密鍵：二つの大きな素数 p と q

公開鍵：$N(=pq), y$

平文　：$m=s_i$　$(0 \leqq m < r)$

暗号文：$E(m)=y^m x^r \mod N$, ただし x は乱数．

多次剰余暗号系では，暗号化に $[2\log_2 r+1]$ 個の mod 演算，および復号化には平均して r 個の mod 演算が必要になる．投票者数を h とすると，$r>h$ であり，h が大きくなると復号化の計算量が増大する．

これに比較して，本方法では，暗号化・復号化ともに $o(10)$ 程度の計算量で十分である．

6. おわりに

一次不定方程式を一方向関数に選んだ暗号系を利用して，ゼロ知識対話証明やマルチパーティプロトコルの実現例を提示し，本方法が有用なことを示した．

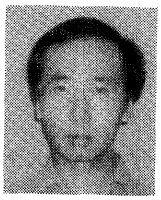
さらに，多次剰余暗号系を利用した場合に比較して本方法の計算量が小さいことも述べた．

今後は，本一方向関数を利用したゼロ知識対話証明を用いて，各種マルチパーティプロトコルの構築を課題としたい．

参考文献

1) Goldwasser, S., Micali, S. and Rackoff, C.: The Knowledge Complexity of Interactive Proof Systems, *Proc. of STOC '85*, pp. 291-304 (1985).
2) 小山謙二：ゼロ知識対話証明の原理と課題，情報処理，Vol. 32, No. 6, pp. 643-653 (1991).
3) 渡辺　治：一方向関数のお話，情報処理，Vol. 32, No. 6, pp. 704-713 (1991).
4) 黒沢　馨，岡本龍明：ゼロ知識証明とマルチパーティプロトコル，情報処理，Vol. 32, No. 6, pp. 663-672 (1991).
5) 八木沢正博：ディオファンタスの一次不定方程式に基づく公開鍵暗号系，情報処理学会論文誌，Vol. 31, No. 12, pp. 1852-1858 (1990).
6) Lenstra, A.K., Lenstra, H.W., Jr. and Lovász, L.: Factoring Polynomials with Rational Coefficients, *Math. Ann.*, 261, pp. 515-534 (1982).

(平成4年6月1日受付)
(平成5年12月9日採録)



八木沢正博（正会員）

昭和25年生．昭和49年東京大学工学部計数工学科卒業．昭和51年同大学院修士課程修了．同年昭和電工(株)入社，川崎工場勤務．昭和61年昭和エンジニアリング(株)に出向，現在に至る．化学プラントの計装エンジニアとして，プラントの設計，保全に従事．現在，素因数分解問題に興味を持つ．