信州大学
大学院工学系研究科

修士論文

2 変数 2 次の多項式補間法による秘密分散法のアクセス構造

指導教員　西新 幹彦 准教授

専攻　電気電子工学専攻
学籍番号　10TA214A
氏名　柏倉 英明

2012 年 3 月 13 日

目次

1 はじめに

秘密情報を漏洩や紛失から守る方法の一つとして秘密分散法がある．これは，いくつかの鍵情報を特定の組み合わせで集めることで秘密情報 S が復元できるというものである．その実現法の一つとして Shamir の多項式補間法があげられる [1] ．この方法では $f(0) = S$ を満たす 1 変数 n 次の多項式からシェアと呼ばれる鍵情報 $v_i = f(i)$ を作り，それらが $n + 1$ 個集まれば多項式が復元出来ることを利用し，秘密情報 $S = f(0)$ を復号する手法である． この方法は集まったシェアの個数により復元の可否が決まるため，しきい値型アクセス構造と呼ばれる．対してシェアの組み合わせで復元の可否が決まるものを一般型アクセス構造と呼ぶ．なお秘密情報が復元できるシェアの組み合わせをアクセス集合，そうでないものを非アクセス集合と呼び，それらの対をアクセス構造と呼ぶ [2].

アクセス構造は完全なものとランプ型に分けられる [3]．非アクセス集合である m 個のシェアが集まったとき，秘密情報のエントロピーが $H(S|v_1, v_2, \cdots, v_m) = H(S)$ となり秘密情報に関する情報が全く漏洩しないものを完全型と呼ぶ．一方で，$H(S|v_1, v_2, \cdots, v_m) < H(S)$ となり，いくらか秘密情報に関する情報が漏洩してしまう場合があるものをランプ型と呼ぶ．このように非アクセス集合の中には，秘密情報が全く漏洩しないものと幾分か漏洩してしまうものの二種類がある．本論文では完全なアクセス構造について論じる．

本研究では上記で述べた多項式補間法を 2 変数に拡張する．1 変数ではシェアの個数のみで復元の可否が決まり，シェアが x 軸上のどの位置にあるかは復号の可否において意味を持たなかった．しかし 2 変数としたとき，シェアは x–y 平面上の位置により各々が特徴付けられ，集まったシェアの個数や位置の組み合わせにより復元の可否が決定するようになる [4]．本研究の目標は，多変数多項式による秘密分散法を考えたとき，実現出来るアクセス構造の全体を明らかにすることである．その目標実現のために，本論文では 2 変数 2 次の場合のアクセス構造を代数的に明らかにする．これにより，実現可能なアクセス構造を全て列挙することが出来る．この成果は，所望のアクセス構造を実現するために，どの位置のシェアを生成すれば良いのかを決める基礎となる．

2 変数の多項式を用いて実現出来るアクセス構造のいくつかは Tassa and Dyn によって発見されている [5]．これらの手法ではまず所望のアクセス構造に対応する多項式を用意する．秘密情報は係数全体の関数となっており，秘密を復元する際に多項式の全係数を復元する必要がある．一方，本論文で提案する手法では秘密情報を多項式の定数項に直接代入して秘密分散を行い，多項式全体を復元せずに秘密が復元出来る場合がある．この点が二つの手法の大きな違いである．

また，本研究は代数幾何符号との関連が強い [6]．本研究では 2 変数の多項式を考えているが，これを射影平面上の有理関数に拡張して考えると，本研究の主題であるアクセス構造を調

べるということは，与えられた複数のシェアがどのような代数曲線上にあるのかを判定し，それに応じた代数幾何符号的な復号条件を調べるということになる．

他の多変数多項式を用いる研究として，多変数公開鍵暗号，秘密関数分散法などが挙げられる [7][8]．多変数公開鍵暗号は公開鍵暗号の実現法の一種で，連立方程式の解を求めることは NP 完全であることに安全性の根拠を求めようとしている暗号方式である．秘密関数分散法は 2 変数の多項式を用いて秘密分散を行うが，アクセス構造はしきい値型となることと，分散される情報が秘密情報ではなく秘密関数となっている点で本研究と異なっている．

実は，任意の一般型完全アクセス構造を実現する方法の一つは既に知られている [9]．しかしこの方法はしきい値型アクセス構造に基づいており，ほとんどのユーザーは複数のシェアを管理する必要がある．ところが 2 変数の多項式補間法を用いると，各ユーザーの管理するシェアを 1 個としたままで一部の一般型アクセス構造を実現出来る [4][5]．

本論文で初めて可能になるアクセス構造の例として，次のようなものがある．所望のアクセス集合として $\{A,B,C\}$, $\{A,B,D,E,F\}$, $\{A,C,D,E,F\}$, $\{B,C,D,E,F\}$ が与えられた場合を考える．しきい値型を基にこれを達成するには，シェアが 9 個あれば秘密情報が復元でき，かつユーザー A,B,C,D,E,F にそれぞれ 3 個，3 個，3 個，1 個，1 個，1 個のシェアを分配すれば良い．この例では A,B,C はそれぞれ一人あたりが D,E,F の三人と同等の重要性を持つ代わりに，D,E,F より多くのシェアを管理する必要がある．言い換えればこの方法では重要なユーザーは多くのシェアを管理しなければならない．しかし 2 変数 2 次の多項式補間法を用いると，各ユーザーが管理するシェアを 1 個としたまま上のアクセス構造が実現出来る．同時に，これは他の関連研究では実現できない例でもある．実現法は 7 節で説明する．

2　2 変数 2 次の多項式補間法

有限体 $\mathbb{F}$ を考える．位数は十分大きな値とする．秘密情報 $S \in \mathbb{F}$ に対し，$\mathbb{F}$ 上の多項式 $f(x,y)$ を

$$f(x,y) = ax^2 + bx + cxy + dy + ey^2 + S \tag{1}$$

と定義する．ここで各係数 a,b,c,d,e はランダムな値を用いる．

シェアは $v_i = (x_i, y_i, z_i)$ といった形で与えられる．$z_i = f(x_i, y_i)$ とし，(x_i, y_i) をシェアの位置と呼ぶ．各ユーザーが管理する非公開情報は z_i のみとし，位置は公開情報とする．したがって各ユーザーが管理する非公開情報のビット数は秘密情報と同じである．このことからユーザーの中で攻撃者に狙われやすい人や狙われにくい人といった差が生じる場合があるが，そうした危険性の大小や対策は本研究の主眼ではないため，本論文では触れないこととする．もし各シェアの位置も非公開情報とすればその問題は解消されるが，各ユーザーが管理する非公開情報の量は 3 倍になる．なお秘密分散の目的から，原点に位置するシェアは生成しない．

秘密情報 S を復元するということは，多項式 $f(x,y)$ の原点における値を求めることに他ならない．集められた複数のシェアから多項式を復元出来れば秘密情報も復元出来るが，多項式を復元しなくても秘密情報を復元出来る場合があるのが，提案法の特徴である．秘密情報を復元するために集めるべきシェアの必要十分条件を次の章で明らかにする．

3 アクセス構造

この章では 2 変数 2 次の多項式補間法により実現できるアクセス構造について述べる．

アクセス集合の一般論として，シェアの集合がアクセス集合を部分集合として含むとき，その集合はアクセス集合となる [9]．よってアクセス集合について論じるときは，極小のアクセス集合について論じれば良い．2 変数 2 次の多項式補間法について考えると，多項式の項が 6 つであることから，極小のアクセス集合のサイズは高々 6 である．従って 7 個以上のシェアが集まったときのアクセス集合は 6 個以下のもののアクセス集合に帰着される．またシェアが 1 個の場合と 2 個の場合，非アクセス集合であることは自明である．以上より，以下の節ではシェアの個数が 3,4,5,6 個の場合のアクセス構造を明らかにする．なおシェアの集合が非アクセス集合なら，そのシェアの部分集合も非アクセス集合であり，証明内で新たにシェアを追加している場合はこのことを用いて証明を進めている．

3.1 3 個のシェアのアクセス構造

定理 1 異なる 3 個のシェア v_1, v_2, v_3 がアクセス集合となるためには，次に示す行列 V_3 の階数が 1 となることが必要十分である．

$$V_3 = \begin{pmatrix} x_1 & y_1 \\ x_2 & y_2 \\ x_3 & y_3 \end{pmatrix} \tag{2}$$

（証明）まず十分性を証明する．式 (2) に示した行列 V_3 の階数が 1 と仮定する．このとき，与えられた 3 個のシェアについて一般性を損なうことなく $y_i = \alpha x_i, (i = 1, 2, 3)$ を満たす α が一意に決定出来る．ここで $x_1 = x_2$ と仮定すると $y_1 = y_2$ が得られ，同じ位置のシェアは与えられないという仮定に矛盾する．このことから x_1, x_2, x_3 は互いに全て異なることが言える．

今シェアの位置と z_i について次の関係が成り立つ．

$$\begin{aligned}
\begin{pmatrix} z_1 \\ z_2 \\ z_3 \end{pmatrix} &= \begin{pmatrix} x_1^2 & x_1 & x_1y_1 & y_1 & y_1^2 & 1 \\ x_2^2 & x_2 & x_2y_2 & y_2 & y_2^2 & 1 \\ x_3^2 & x_3 & x_3y_3 & y_3 & y_3^2 & 1 \end{pmatrix} \begin{pmatrix} a \\ b \\ c \\ d \\ e \\ S \end{pmatrix} \\
&= \begin{pmatrix} x_1^2 & x_1 & \alpha x_1^2 & \alpha x_1 & \alpha^2 x_1^2 & 1 \\ x_2^2 & x_2 & \alpha x_2^2 & \alpha x_2 & \alpha^2 x_2^2 & 1 \\ x_3^2 & x_3 & \alpha x_3^2 & \alpha x_3 & \alpha^2 x_3^2 & 1 \end{pmatrix} \begin{pmatrix} a \\ b \\ c \\ d \\ e \\ S \end{pmatrix} \\
&= \begin{pmatrix} x_1^2 & x_1 & 1 \\ x_2^2 & x_2 & 1 \\ x_3^2 & x_3 & 1 \end{pmatrix} \begin{pmatrix} a + \alpha c + \alpha^2 e \\ b + \alpha d \\ S \end{pmatrix} \qquad (3)
\end{aligned}$$

式 (3) の右辺の正方行列は Vandermonde の行列式より正則であり，逆行列が存在する [10]．この逆行列を両辺に左からかけることで S の値を得ることが出来るため，このとき秘密情報が復元出来る．

次に必要性を証明する．原点に位置するシェアは存在しないという仮定から，行列 V_3 の階数は 0 ではない．よって V_3 の階数を 2 と仮定して，秘密情報が復元できないことを示す．一般性を損なうことなく $x_1 \neq 0$，かつある α に対して

$$y_1 = \alpha x_1, \quad y_2 \neq \alpha x_2, \quad y_3 \neq \alpha x_3 \qquad (4)$$

が成り立つと仮定する ($\alpha \neq 0$ または $\alpha = 0$)．ここで

$$(x_4, y_4) \neq (x_1, y_1) \qquad (5)$$

$$y_4 = \alpha x_4 \qquad (6)$$

$$y_5 \neq \alpha x_5 \qquad (7)$$

$$\begin{vmatrix} x_2 - x_5 & y_2 - y_5 \\ x_3 - x_5 & y_3 - y_5 \end{vmatrix} \neq 0 \qquad (8)$$

を満たす位置に，シェア v_4, v_5 を考える[*1]．また z_4, z_5 は任意の値とする．$\{v_1, v_2, v_3, v_4, v_5\}$ が非アクセス集合なら $\{v_1, v_2, v_3\}$ は非アクセス集合であることから必要性を示す．ここで秘密情報の値について考える．集まったシェアに対して，秘密情報の値として取り得ない値があれば，秘密情報の値の候補が減り，秘密情報のエントロピー $H(S|v_1, v_2 \cdots, v_5)$ が $H(S)$ より

[*1] $\mathbb{F}$ のサイズを 7 以上とすると，このような位置を考えることが出来る．証明は付録に示す．以降，同様の注意は省略する．

小さくなることになる．よってどんな値 $\hat{S}$ を秘密情報として仮定しても集まったシェアと矛盾しないことを示せば必要性が言える．そこで秘密情報が $\hat{S}$ であると仮定すると，

$$\begin{pmatrix} z_1 - \hat{S} \\ z_4 - \hat{S} \end{pmatrix} = \begin{pmatrix} x_1^2 & x_1 & x_1y_1 & y_1 & y_1^2 \\ x_4^2 & x_4 & x_4y_4 & y_4 & y_4^2 \end{pmatrix} \begin{pmatrix} a \\ b \\ c \\ d \\ e \end{pmatrix} \tag{9}$$

が成り立たなければならない．ここで右辺の 2×5 行列に着目すると，式 (4)，式 (6) より

$$\begin{aligned} &\begin{pmatrix} x_1^2 & x_1 & x_1y_1 & y_1 & y_1^2 \\ x_4^2 & x_4 & x_4y_4 & y_4 & y_4^2 \end{pmatrix} \\ &= \begin{pmatrix} x_1^2 & x_1 & \alpha x_1^2 & \alpha x_1 & \alpha^2 x_1^2 \\ x_4^2 & x_4 & \alpha x_4^2 & \alpha x_4 & \alpha^2 x_4^2 \end{pmatrix} \\ &= \begin{pmatrix} x_1^2 & x_1 \\ x_4^2 & x_4 \end{pmatrix} \begin{pmatrix} 1 & 0 & \alpha & 0 & \alpha^2 \\ 0 & 1 & 0 & \alpha & 0 \end{pmatrix} \end{aligned} \tag{10}$$

と書ける．上式の正方行列は Vandermonde の行列式より正則である．この逆行列を式 (9) の両辺に左からかけると

$$\begin{pmatrix} A \\ B \end{pmatrix} = \begin{pmatrix} 1 & 0 & \alpha & 0 & \alpha^2 \\ 0 & 1 & 0 & \alpha & 0 \end{pmatrix} \begin{pmatrix} a \\ b \\ c \\ d \\ e \end{pmatrix} \tag{11}$$

を満たす A，B の値を求めることが出来る．ここで式 (11) と z_2，z_3，z_5 から次式を得る．

$$\begin{pmatrix} A \\ B \\ z_2 - \hat{S} \\ z_3 - \hat{S} \\ z_5 - \hat{S} \end{pmatrix} = \begin{pmatrix} 1 & 0 & \alpha & 0 & \alpha^2 \\ 0 & 1 & 0 & \alpha & 0 \\ x_2^2 & x_2 & x_2y_2 & y_2 & y_2^2 \\ x_3^2 & x_3 & x_3y_3 & y_3 & y_3^2 \\ x_5^2 & x_5 & x_5y_5 & y_5 & y_5^2 \end{pmatrix} \begin{pmatrix} a \\ b \\ c \\ d \\ e \end{pmatrix} \tag{12}$$

右辺の行列式の値を計算すると

$$\begin{aligned} &\begin{vmatrix} 1 & 0 & \alpha & 0 & \alpha^2 \\ 0 & 1 & 0 & \alpha & 0 \\ x_2^2 & x_2 & x_2y_2 & y_2 & y_2^2 \\ x_3^2 & x_3 & x_3y_3 & y_3 & y_3^2 \\ x_5^2 & x_5 & x_5y_5 & y_5 & y_5^2 \end{vmatrix} \\ &= (\alpha x_2 - y_2)(\alpha x_3 - y_3)(\alpha x_5 - y_5)\{(x_2 - x_5)(y_3 - y_5) - (x_3 - x_5)(y_2 - y_5)\} \end{aligned} \tag{13}$$

となり，式 (4)，式 (7)，式 (8) より式 (13) は非零の値を持つ．よって式 (12) の右辺の正方行列は逆行列が存在するため，今考えている v_4，v_5，$\hat{S}$ について，それらを満たす a，b，c，d，e が存在する．

以上より，各 $\hat{S}$ に対して秘密情報が $\hat{S}$ であるという仮定をおくと，それらの仮定は同様に確からしいため，真の秘密情報 S に関する情報は得られない．□

3.2 4 個のシェアのアクセス構造

定理 2 異なる 4 個のシェアがアクセス集合であるための必要十分条件は，ある真部分集合がアクセス集合であることである．

（証明）十分性は自明であるので，必要性を示す．異なる 4 個のシェア v_1, v_2, v_3, v_4 からなる集合を考える．一般性を損なうことなく，任意の真部分集合が非アクセス集合であると仮定する．今ある 4 個のシェアに新たにシェア v_5 を加えた場合を考える．定理 1 の必要性の証明と同様に，秘密情報の値をどの $\hat{S}$ と仮定しても，5 個のシェアの位置と $\hat{S}$ の値に矛盾しない多項式が一意に決定されることを示す．すると $\{v_1, v_2, v_3, v_4, v_5\}$ が非アクセス集合となるので，$\{v_1, v_2, v_3, v_4\}$ もまた非アクセス集合となることから必要性が示される．

まず，2 つのシェア v_i, v_j に対して

$$x_i y_j - x_j y_i = 0 \tag{14}$$

によって同値関係を定義し，この同値関係を用いてシェアの位置に関する同値類を定義する．今どの 3 個を選んでも非アクセス集合なので，定理 1 より，どの同値類も高々 2 個のシェアからなる．

まず 2 個のシェアからなる同値類がある場合について考える．すなわち一般性を損なうことなく v_1 と v_2 が同値であると仮定する．次式を満たす任意の位置にシェア v_5 を選ぶ．

$$\begin{vmatrix} x_4 - x_3 & y_4 - y_3 \\ x_5 - x_3 & y_5 - y_3 \end{vmatrix} \neq 0 \tag{15}$$

$$x_1 y_5 - x_5 y_1 \neq 0 \tag{16}$$

なお z_5 の値は任意の値とする．

以上より，今 5 個のシェアが与えられ，それらは次式を満たす．

$$x_1 y_2 - x_2 y_1 = 0 \tag{17}$$

$$\begin{vmatrix} x_4 - x_3 & y_4 - y_3 \\ x_5 - x_3 & y_5 - y_3 \end{vmatrix} \neq 0 \tag{18}$$

$$x_i y_j - x_j y_i \neq 0 \ (i = 1, 2 \ , j = 3, 4, 5) \tag{19}$$

一般性を損なうことなく $x_1 \neq 0$ と仮定する．もし $x_2 = 0$ なら，式 (17) にて，原点のシェアは与えられないことから $x_1 = 0$ となる．したがって，$x_1 \neq 0$ なら $x_2 \neq 0$ である．ここで秘

密情報を任意の $\hat{S}$ と仮定すると

$$\begin{pmatrix} z_1 - \hat{S} \\ z_2 - \hat{S} \\ z_3 - \hat{S} \\ z_4 - \hat{S} \\ z_5 - \hat{S} \end{pmatrix} = \begin{pmatrix} x_1^2 & x_1 & x_1y_1 & y_1 & y_1^2 \\ x_2^2 & x_2 & x_2y_2 & y_2 & y_2^2 \\ x_3^2 & x_3 & x_3y_3 & y_3 & y_3^2 \\ x_4^2 & x_4 & x_4y_4 & y_4 & y_4^2 \\ x_5^2 & x_5 & x_5y_5 & y_5 & y_5^2 \end{pmatrix} \begin{pmatrix} a \\ b \\ c \\ d \\ e \end{pmatrix} \tag{20}$$

が成り立たなければならない．右辺の正方行列が正則であれば，任意の $\hat{S}$ を考えたとき，それを満たす多項式の各係数が一意に決定出来ることが言える．正方行列の行列式の値

$$\begin{vmatrix} x_1^2 & x_1 & x_1y_1 & y_1 & y_1^2 \\ x_2^2 & x_2 & x_2y_2 & y_2 & y_2^2 \\ x_3^2 & x_3 & x_3y_3 & y_3 & y_3^2 \\ x_4^2 & x_4 & x_4y_4 & y_4 & y_4^2 \\ x_5^2 & x_5 & x_5y_5 & y_5 & y_5^2 \end{vmatrix} \tag{21}$$

を計算する．式 (21) にて列操作をすることで次式を得る．ただし $\alpha = y_1/x_1$ とした．

$$\begin{aligned} &\begin{vmatrix} x_1^2 & x_1 & x_1y_1 - \alpha x_1^2 & y_1 - \alpha x_1 & y_1^2 - \alpha^2 x_1^2 \\ x_2^2 & x_2 & x_2y_2 - \alpha x_2^2 & y_2 - \alpha x_2 & y_2^2 - \alpha^2 x_2^2 \\ x_3^2 & x_3 & x_3y_3 - \alpha x_3^2 & y_3 - \alpha x_3 & y_3^2 - \alpha^2 x_3^2 \\ x_4^2 & x_4 & x_4y_4 - \alpha x_4^2 & y_4 - \alpha x_4 & y_4^2 - \alpha^2 x_4^2 \\ x_5^2 & x_5 & x_5y_5 - \alpha x_5^2 & y_5 - \alpha x_5 & y_5^2 - \alpha^2 x_5^2 \end{vmatrix} \\ = &\begin{vmatrix} x_1^2 & x_1 & 0 & 0 & 0 \\ x_2^2 & x_2 & 0 & 0 & 0 \\ x_3^2 & x_3 & x_3y_3 - \alpha x_3^2 & y_3 - \alpha x_3 & y_3^2 - \alpha^2 x_3^2 \\ x_4^2 & x_4 & x_4y_4 - \alpha x_4^2 & y_4 - \alpha x_4 & y_4^2 - \alpha^2 x_4^2 \\ x_5^2 & x_5 & x_5y_5 - \alpha x_5^2 & y_5 - \alpha x_5 & y_5^2 - \alpha^2 x_5^2 \end{vmatrix} \\ &= x_1x_2(x_2 - x_1)(y_3 - \alpha x_3)(y_4 - \alpha x_4)(y_5 - \alpha x_5) \\ &\cdot \{(x_4 - x_3)(y_5 - y_3) - (x_5 - x_3)(y_4 - y_3)\} \end{aligned} \tag{22}$$

$x_1 = x_2$ を仮定すると，式 (17) より $y_1 = y_2$ が得られるが，それはシェアの位置は異なるという仮定と矛盾する．よって今 $x_2 - x_1 \neq 0$ が言える．また式 (18)，式 (19) より式 (22) は非零の因数の積なので，式 (20) の右辺の行列は正則である．よって任意の秘密情報 $\hat{S}$ を考えたとき，それを満たす多項式の各係数が一意に決定出来る．

以上より，各 $\hat{S}$ に対して，秘密情報の値が $\hat{S}$ であるという仮定をおくと，それらの仮定は同様に確からしいため，真の秘密情報 S に関する情報は得られない．

次に，2 個のシェアからなる同値類がない場合について考える．つまりこのときどの 2 個を選んでも同値関係でない．ある任意の 3 個について考えたとき，次式を満たす組み合わせがある場合とない場合で更に場合分けをする．

$$\begin{vmatrix} x_1 - x_3 & y_1 - y_3 \\ x_2 - x_3 & y_2 - y_3 \end{vmatrix} \neq 0 \tag{23}$$

まず式 (23) を満たす 3 個がある場合について考える．このとき次式を満たす任意の位置にシェア v_5 を選ぶ．

$$x_4y_5 - x_5y_4 = 0 \tag{24}$$

なお z_5 の値は任意とする．ここでシェア v_1, v_2, v_3, v_4, v_5 の名前をそれぞれ v_3, v_4, v_5, v_1, v_2 につけかえれば証明は式 (17) 以降の手順に帰着する．

次に，どの 3 個を選んでも式 (23) が満たされない場合を考える．このとき

$$y_3 = \frac{x_3y_2 - x_1y_2 - x_3y_1 + x_2y_1}{x_2 - x_1} \tag{25}$$

が成り立つ．また

$$x_4y_5 - x_5y_4 = 0 \tag{26}$$

を満たす位置に，任意の z_5 の値を持つシェア v_5 を選ぶ．すると，後述の定理 3 よりこの 5 個のシェアが非アクセス集合であることが次のようにして言える．このとき式 (28) で定める行列 V_5 に式 (25) を代入すると，$|V_5| = 0$ となる．よって後述の補題 1 より今行列 V_5 の階数は 3 か 4 である．そして行列 V_5 のどの列要素を全て 1 にした行列 V_5' の行列式を考えても，零の因数

$$\begin{vmatrix} x_1 - x_4 & y_1 - y_4 \\ x_2 - x_4 & y_2 - y_4 \end{vmatrix} = 0 \tag{27}$$

を持つので $|V_5'| = 0$ となる．よって定理 3 より非アクセス集合であることが言える．任意の z_5 を持つシェア v_5 を与えても秘密を復元出来ないので，今ある 4 個のシェアは非アクセス集合である．□

この定理は，2 変数 2 次の多項式補間法では極小のサイズが 4 となるアクセス集合は存在しないことを意味する．

なお定理 2 の証明で定理 3 を用いたが，定理 3 の証明では定理 2 を用いていないので，論理の循環は起きていない．

3.3 5 個のシェアのアクセス構造

定理 3 異なる 5 個のシェア v_1, v_2, v_3, v_4, v_5 がアクセス集合であるための必要十分条件は，ある真部分集合がアクセス集合であるか，または以下に定める行列 V_5 の階数が 4 でありかつ V_5 のある列の要素をすべて 1 にした行列 V_5' の階数が 5 となることである．

$$V_5 = \begin{pmatrix} x_1^2 & x_1 & x_1y_1 & y_1 & y_1^2 \\ x_2^2 & x_2 & x_2y_2 & y_2 & y_2^2 \\ x_3^2 & x_3 & x_3y_3 & y_3 & y_3^2 \\ x_4^2 & x_4 & x_4y_4 & y_4 & y_4^2 \\ x_5^2 & x_5 & x_5y_5 & y_5 & y_5^2 \end{pmatrix} \tag{28}$$

（証明）ある真部分集合がアクセス集合である場合は自明なので，以降は任意の真部分集合が非アクセス集合であると仮定して証明を進める．

十分性を示す．式 (28) に示す行列 V_5 の階数が 4 と仮定する．このとき行列 V_5 のうち 1 列は，他の 4 列に線形従属である．また

$$\begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} = V_5 \begin{pmatrix} A \\ B \\ C \\ D \\ E \end{pmatrix} \tag{29}$$

は自明でない解 A，B，C，D，E を持つ．一般性を損なうことなく $E = -1$ とする．ここで 5 個のシェアの位置について

$$y^2 = Ax^2 + Bx + Cxy + Dy \tag{30}$$

が成り立つ．式 (30) より多項式 $f(x, y)$ は次式を満たす．

$$\begin{aligned} f(x,y) &= ax^2 + bx + cxy + dy + ey^2 + S \\ &= (a + Ae)x^2 + (b + Be)x + (c + Ce)xy + (d + De)y + S \\ &= a'x^2 + b'x + c'xy + d'y + S \end{aligned} \tag{31}$$

式 (31) より，次の式が得られる．

$$\begin{pmatrix} z_1 \\ z_2 \\ z_3 \\ z_4 \\ z_5 \end{pmatrix} = \begin{pmatrix} x_1^2 & x_1 & x_1y_1 & y_1 & 1 \\ x_2^2 & x_2 & x_2y_2 & y_2 & 1 \\ x_3^2 & x_3 & x_3y_3 & y_3 & 1 \\ x_4^2 & x_4 & x_4y_4 & y_4 & 1 \\ x_5^2 & x_5 & x_5y_5 & y_5 & 1 \end{pmatrix} \begin{pmatrix} a' \\ b' \\ c' \\ d' \\ S \end{pmatrix} \tag{32}$$

式 (32) の右辺の正方行列を行列 V_5' とする．ここで行列 V_5' の階数が 5 と仮定すると，行列 V_5' の逆行列を両辺に左からかけることで秘密情報 S が復元出来る．

次に必要性を示す．仮定と次の補題 1 より，行列 V_5 の階数が 3，5，または 4 かつ V_5' の階数が 4 の場合について，それぞれ秘密情報がどんな値であると仮定しても，それらの仮定が同様に確からしいことから必要性を示す．

1. 行列 V_5 の階数が 5 の場合

任意の $\hat{S}$ について次の式を考える．

$$\begin{pmatrix} z_1 - \hat{S} \\ z_2 - \hat{S} \\ z_3 - \hat{S} \\ z_4 - \hat{S} \\ z_5 - \hat{S} \end{pmatrix} = V_5 \begin{pmatrix} a \\ b \\ c \\ d \\ e \end{pmatrix} \tag{33}$$

行列 V_5 の階数は 5 なので，今任意の $\hat{S}$ に対して式 (33) を満たす a，b，c，d，e が存在し，また一意に決定される．

以上より，どのような $\hat{S}$ を秘密情報と仮定しても，それを満たす多項式が同様に確からしく存在することが言えたので，今秘密情報 S に関する情報は得られない．

2. 行列 V_5 の階数が 4，行列 V_5' の階数が 4 の場合

このとき行列 V_5 のうち 1 列は，他の 4 列に線形従属である．また

$$\begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} = V_5 \begin{pmatrix} A \\ B \\ C \\ D \\ E \end{pmatrix} \tag{34}$$

は自明でない解 A，B，C，D，E を持つ．一般性を損なうことなく $E = -1$ とする．このとき 5 列目は他の 4 列に線形従属である．ここで 5 個のシェアの位置について，次式が成り立つ．

$$y^2 = Ax^2 + Bx + Cxy + Dy \tag{35}$$

式 (35) より多項式 $f(x, y)$ は次式を満たす．

$$\begin{aligned} f(x,y) &= ax^2 + bx + cxy + dy + ey^2 + S \\ &= (a + Ae)x^2 + (b + Be)x + (c + Ce)xy + (d + De)y + S \\ &= a'x^2 + b'x + c'xy + d'y + S \end{aligned} \tag{36}$$

式 (36) より，次の式が得られる．

$$\begin{pmatrix} z_1 \\ z_2 \\ z_3 \\ z_4 \\ z_5 \end{pmatrix} = \begin{pmatrix} x_1^2 & x_1 & x_1y_1 & y_1 & 1 \\ x_2^2 & x_2 & x_2y_2 & y_2 & 1 \\ x_3^2 & x_3 & x_3y_3 & y_3 & 1 \\ x_4^2 & x_4 & x_4y_4 & y_4 & 1 \\ x_5^2 & x_5 & x_5y_5 & y_5 & 1 \end{pmatrix} \begin{pmatrix} a' \\ b' \\ c' \\ d' \\ S \end{pmatrix} \tag{37}$$

式 (37) の右辺の正方行列を V_5' とおく．これは行列 V_5 のうち他の 4 列に線形従属であった 5 列目の要素を全て 1 にしたものである．このとき V_5' の部分行列を考えると，階数が 4 となるものが存在する．一般性を損なうことなく次に示す行列 V_4 の階数を 4 とする．

$$V_4 = \begin{pmatrix} x_1^2 & x_1 & x_1y_1 & y_1 \\ x_2^2 & x_2 & x_2y_2 & y_2 \\ x_3^2 & x_3 & x_3y_3 & y_3 \\ x_4^2 & x_4 & x_4y_4 & y_4 \end{pmatrix} \tag{38}$$

ここで任意の $\hat{S}$ に対して

$$\begin{pmatrix} A' \\ B' \\ C' \\ D' \end{pmatrix} \triangleq V_4^{-1} \begin{pmatrix} z_1 - \hat{S} \\ z_2 - \hat{S} \\ z_3 - \hat{S} \\ z_4 - \hat{S} \end{pmatrix} \tag{39}$$

のように A', B', C', D' を定めると多項式

$$f'(x, y) = A'x^2 + B'x + C'xy + D'y + \hat{S} \tag{40}$$

はシェア v_1, v_2, v_3, v_4 の値と矛盾しない．さらにこれがシェア v_5 の値とも矛盾しないことが以下のように示される．V_4 の階数が 4 であることから，式 (37) より今ある α_1，α_2，α_3，α_4 が存在し，それらは次式を満たす．

$$\begin{aligned} &\begin{pmatrix} x_5^2 & x_5 & x_5y_5 & y_5 & 1 \end{pmatrix} \\ &\quad = \begin{pmatrix} \alpha_1 & \alpha_2 & \alpha_3 & \alpha_4 \end{pmatrix} V_4' \end{aligned} \tag{41}$$

$$z_5 = \begin{pmatrix} \alpha_1 & \alpha_2 & \alpha_3 & \alpha_4 \end{pmatrix} \begin{pmatrix} z_1 \\ z_2 \\ z_3 \\ z_4 \end{pmatrix} \tag{42}$$

ただし

$$V_4' = \begin{pmatrix} x_1^2 & x_1 & x_1y_1 & y_1 & 1 \\ x_2^2 & x_2 & x_2y_2 & y_2 & 1 \\ x_3^2 & x_3 & x_3y_3 & y_3 & 1 \\ x_4^2 & x_4 & x_4y_4 & y_4 & 1 \end{pmatrix} \tag{43}$$

とした．ここで $f'(x_5, y_5)$ を考える．

$$\begin{aligned} &f'(x_5, y_5) \\ &= \begin{pmatrix} x_5^2 & x_5 & x_5y_5 & y_5 \end{pmatrix} \begin{pmatrix} A' \\ B' \\ C' \\ D' \end{pmatrix} + \hat{S} \\ &= \begin{pmatrix} \alpha_1 & \alpha_2 & \alpha_3 & \alpha_4 \end{pmatrix} V_4 V_4^{-1} \begin{pmatrix} z_1 - \hat{S} \\ z_2 - \hat{S} \\ z_3 - \hat{S} \\ z_4 - \hat{S} \end{pmatrix} + \hat{S} \\ &= \begin{pmatrix} \alpha_1 & \alpha_2 & \alpha_3 & \alpha_4 \end{pmatrix} \begin{pmatrix} z_1 - \hat{S} \\ z_2 - \hat{S} \\ z_3 - \hat{S} \\ z_4 - \hat{S} \end{pmatrix} + \hat{S} \\ &= z_5 \end{aligned} \tag{44}$$

以上より，多項式 $f'(x,y)$ は v_5 に矛盾せず成立することが言えた．よってこのとき秘密情報を任意の $\hat{S}$ と仮定すると，ある 4 個のシェアからそれを満たす多項式が一意に決定でき，またそれは残った 1 個のシェアについても矛盾せず成立する．どのような $\hat{S}$ を秘密情報と仮定しても，それらの仮定は同様に確からしいため，秘密情報 S に関する情報は得られない．

3. 行列 V_5 の階数が 3 の場合

このとき行列 V_5 の部分行列を考えると，階数が 3 となるものが存在する．一般性を損なうことなく次に示す行列 V_3 の階数は 3 とする．

$$V_3 = \begin{pmatrix} x_1y_1 & y_1 & y_1^2 \\ x_2y_2 & y_2 & y_2^2 \\ x_3y_3 & y_3 & y_3^2 \end{pmatrix} \tag{45}$$

ここで任意の $\hat{S}$，A，B に対して

$$\begin{pmatrix} C \\ D \\ E \end{pmatrix} \triangleq V_3^{-1} \begin{pmatrix} z_1 - \hat{S} - Ax_1^2 - Bx_1 \\ z_2 - \hat{S} - Ax_2^2 - Bx_2 \\ z_3 - \hat{S} - Ax_3^2 - Bx_3 \end{pmatrix} \tag{46}$$

のように C，D，E を定めると多項式

$$f'(x,y) = Ax^2 + Bx + Cxy + Dy + Ey^2 + \hat{S} \tag{47}$$

はシェア v_1，v_2，v_3 の値と矛盾しない．さらにこれがシェア v_4，v_5 の値とも矛盾しないことが以下のように示される．

V_3 の階数が 3 であることから，ある α_1，α_2，α_3，β_1，β_2，β_3 が存在し，それらは次式を満たす．

$$\begin{aligned} &\begin{pmatrix} x_4^2 & x_4 & x_4y_4 & y_4 & y_4^2 \\ x_5^2 & x_5 & x_5y_5 & y_5 & y_5^2 \end{pmatrix} \\ &= \begin{pmatrix} \alpha_1 & \alpha_2 & \alpha_3 \\ \beta_1 & \beta_2 & \beta_3 \end{pmatrix} \begin{pmatrix} x_1^2 & x_1 & x_1y_1 & y_1 & y_1^2 \\ x_2^2 & x_2 & x_2y_2 & y_2 & y_2^2 \\ x_3^2 & x_3 & x_3y_3 & y_3 & y_3^2 \end{pmatrix} \end{aligned} \tag{48}$$

$$\begin{pmatrix} z_4 - \hat{S} \\ z_5 - \hat{S} \end{pmatrix} = \begin{pmatrix} \alpha_1 & \alpha_2 & \alpha_3 \\ \beta_1 & \beta_2 & \beta_3 \end{pmatrix} \begin{pmatrix} z_1 - \hat{S} \\ z_2 - \hat{S} \\ z_3 - \hat{S} \end{pmatrix} \tag{49}$$

ここで $f'(x_4, y_4)$，$f'(x_5, y_5)$ を考える．

$$
\begin{aligned}
&\begin{pmatrix} f'(x_4, y_4) \\ f'(x_5, y_5) \end{pmatrix} \\
&= \begin{pmatrix} x_4 y_4 & y_4 & y_4^2 \\ x_5 y_5 & y_5 & y_5^2 \end{pmatrix} \begin{pmatrix} C \\ D \\ E \end{pmatrix} + \begin{pmatrix} Ax_4^2 + Bx_4 + \hat{S} \\ Ax_5^2 + Bx_5 + \hat{S} \end{pmatrix} \\
&= \begin{pmatrix} \alpha_1 & \alpha_2 & \alpha_3 \\ \beta_1 & \beta_2 & \beta_3 \end{pmatrix} V_3 V_3^{-1} \begin{pmatrix} z_1 - \hat{S} - Ax_1^2 - Bx_1 \\ z_2 - \hat{S} - Ax_2^2 - Bx_2 \\ z_3 - \hat{S} - Ax_3^2 - Bx_3 \end{pmatrix} \\
&\quad + \begin{pmatrix} Ax_4^2 + Bx_4 + \hat{S} \\ Ax_5^2 + Bx_5 + \hat{S} \end{pmatrix} \\
&= \begin{pmatrix} \alpha_1 & \alpha_2 & \alpha_3 \\ \beta_1 & \beta_2 & \beta_3 \end{pmatrix} \begin{pmatrix} z_1 - \hat{S} - Ax_1^2 - Bx_1 \\ z_2 - \hat{S} - Ax_2^2 - Bx_2 \\ z_3 - \hat{S} - Ax_3^2 - Bx_3 \end{pmatrix} \\
&\quad + \begin{pmatrix} Ax_4^2 + Bx_4 + \hat{S} \\ Ax_5^2 + Bx_5 + \hat{S} \end{pmatrix} \\
&= \begin{pmatrix} z_4 - \hat{S} - Ax_4^2 - Bx_4 \\ z_5 - \hat{S} - Ax_5^2 - Bx_5 \end{pmatrix} + \begin{pmatrix} Ax_4^2 + Bx_4 + \hat{S} \\ Ax_5^2 + Bx_5 + \hat{S} \end{pmatrix} \\
&= \begin{pmatrix} z_4 \\ z_5 \end{pmatrix}
\end{aligned} \tag{50}
$$

以上より，多項式 $f'(x, y)$ は v_4，v_5 に矛盾せず成立することが言えた．よってこのとき $\hat{S}$ と多項式の 2 係数を任意に定めると，ある 3 個のシェアからそれを満たす多項式が一意に決定でき，またそれは残った 2 個のシェアについても矛盾せず成立する．秘密情報を任意の $\hat{S}$ と仮定しても，それらの仮定は同様に確からしいため，秘密情報 S に関する情報は得られない．□

補題 1　5 個のシェアのアクセス構造について考えるとき，行列 V_5 の階数が 2 以下なら，どの 3 個を選んでもアクセス集合である．

（証明）5 個の異なるシェアのうち，任意の 3 個が非アクセス集合なら，行列 V_5 の階数は少なくとも 3 であることを示し，対偶より補題を導く．まず与えられた 5 個のシェアの位置について式 (14) の同値関係を考える．同値な 2 個のシェアがあるかどうかで場合分けをし，それぞれの場合について行列 V_5 の部分行列を考える．

1. どの 2 個も同値でない場合

行列 V_5 の 1 列目，3 列目，5 列目と 1 行目，2 行目，3 行目についての部分行列は次のよう

になる．

$$\begin{pmatrix} x_1^2 & x_1y_1 & y_1^2 \\ x_2^2 & x_2y_2 & y_2^2 \\ x_3^2 & x_3y_3 & y_3^2 \end{pmatrix} \tag{51}$$

また式 (51) の行列式の値は

$$\begin{vmatrix} x_1^2 & x_1y_1 & y_1^2 \\ x_2^2 & x_2y_2 & y_2^2 \\ x_3^2 & x_3y_3 & y_3^2 \end{vmatrix} \\ = (x_1y_2 - x_2y_1)(x_2y_3 - x_3y_2)(x_1y_3 - x_3y_1) \tag{52}$$

である．仮定より式 (52) の右辺の値は非零の因数の積なので非零であり，式 (51) に示す行列の階数は 3 である．よってこのとき，行列 V_5 の階数は少なくとも 3 である．

2. 同値の 2 個がある場合

一般性を損なうことなく

$$x_1y_2 - x_2y_1 = 0 \tag{53}$$

とする．すると

$$x_1y_3 - x_3y_1 \neq 0 \tag{54}$$

となる．何故なら式 (54) でないとすると，定理 1 より $\{v_1, v_2, v_3\}$ がアクセス集合になり，仮定に反するからである．さらに，一般性を損なうことなく $x_1 \neq 0$ と仮定する．すると式 (53) より $x_2 \neq 0$ かつ $x_1 \neq x_2$ となる．

ここで次のような部分行列を考える．

$$\begin{pmatrix} x_1^2 & x_1 & y_1 \\ x_2^2 & x_2 & y_2 \\ x_3^2 & x_3 & y_3 \end{pmatrix} \tag{55}$$

2 列目を用いて 3 列目の 1 行目と 2 行目の要素を消去すると次のようになる．

$$\begin{pmatrix} x_1^2 & x_1 & 0 \\ x_2^2 & x_2 & 0 \\ x_3^2 & x_3 & \frac{x_1y_3 - x_3y_1}{x_1} \end{pmatrix} \tag{56}$$

式 (54) より，式 (54) の 3 行 3 列の要素は必ず非零となる．続いて 3 列目を用いて 1 列目と 2 列目の 3 行目を消去すると，式 (56) はある非零の値 y_3' を用いて次のように書ける．

$$\begin{pmatrix} x_1^2 & x_1 & 0 \\ x_2^2 & x_2 & 0 \\ 0 & 0 & y_3' \end{pmatrix} \tag{57}$$

$x_1 \neq x_2$ より上の行列の階数は 3 なので，このとき行列 V_5 の階数は少なくとも 3 である．□

3.4　6 個のシェアのアクセス構造

定理 4　異なる 6 個のシェア $v_1, v_2, v_3, v_4, v_5, v_6$ がアクセス集合であるための必要十分条件は，ある真部分集合がアクセス集合であるか，または以下に定める行列 V_6 の階数が 6 となることである．

$$V_6 = \begin{pmatrix} x_1^2 & x_1 & x_1y_1 & y_1 & y_1^2 & 1 \\ x_2^2 & x_2 & x_2y_2 & y_2 & y_2^2 & 1 \\ x_3^2 & x_3 & x_3y_3 & y_3 & y_3^2 & 1 \\ x_4^2 & x_4 & x_4y_4 & y_4 & y_4^2 & 1 \\ x_5^2 & x_5 & x_5y_5 & y_5 & y_5^2 & 1 \\ x_6^2 & x_6 & x_6y_6 & y_6 & y_6^2 & 1 \end{pmatrix} \tag{58}$$

（証明）ある真部分集合がアクセス集合である場合は自明なので，以降は任意の真部分集合が非アクセス集合であると仮定して証明を進める．

まず十分性を証明する．行列 V_6 の階数を 6 と仮定する．このとき次式が成り立つ．

$$\begin{pmatrix} z_1 \\ z_2 \\ z_3 \\ z_4 \\ z_5 \\ z_6 \end{pmatrix} = V_6 \begin{pmatrix} a \\ b \\ c \\ d \\ e \\ S \end{pmatrix} \tag{59}$$

ここで行列 V_6 は逆行列が存在し，それを両辺に左からかけることで多項式の各係数が復元できる．よってこのとき秘密情報 S が復元可能である．

次に必要性を示す．定理 3 同様，全ての場合において，どのような秘密情報 $\hat{S}$ を秘密情報と仮定しても，それらが同様に確からしいことを示す．行列 V_6 の階数が 1 の場合から 5 の場合まで考える必要があるが，仮定と補題 1 より，行列 V_6 の階数が 3，4，5 の場合について証明すれば良い．

1. V_6 の階数が 5 の場合

一般性を損なうことなく，V_6 にて 6 行目は他の 5 行に線形従属であると仮定する．このとき，ある α_1，α_2，α_3，α_4，α_5 が存在し，次式を満たす．

$$\begin{aligned} &\begin{pmatrix} x_6^2 & x_6 & x_6y_6 & y_6 & y_6^2 & 1 \end{pmatrix} \\ &= \begin{pmatrix} \alpha_1 & \alpha_2 & \alpha_3 & \alpha_4 & \alpha_5 \end{pmatrix} V_6' \end{aligned} \tag{60}$$

$$z_6 = \begin{pmatrix} \alpha_1 & \alpha_2 & \alpha_3 & \alpha_4 & \alpha_5 \end{pmatrix} \begin{pmatrix} z_1 \\ z_2 \\ z_3 \\ z_4 \\ z_5 \end{pmatrix} \tag{61}$$

ただし

$$V_6' = \begin{pmatrix} x_1^2 & x_1 & x_1y_1 & y_1 & y_1^2 & 1 \\ x_2^2 & x_2 & x_2y_2 & y_2 & y_2^2 & 1 \\ x_3^2 & x_3 & x_3y_3 & y_3 & y_3^2 & 1 \\ x_4^2 & x_4 & x_4y_4 & y_4 & y_4^2 & 1 \\ x_5^2 & x_5 & x_5y_5 & y_5 & y_5^2 & 1 \end{pmatrix} \tag{62}$$

とした.

ここで次のような行列 V_5 を考え，その階数について場合分けをする.

$$V_5 = \begin{pmatrix} x_1^2 & x_1 & x_1y_1 & y_1 & y_1^2 \\ x_2^2 & x_2 & x_2y_2 & y_2 & y_2^2 \\ x_3^2 & x_3 & x_3y_3 & y_3 & y_3^2 \\ x_4^2 & x_4 & x_4y_4 & y_4 & y_4^2 \\ x_5^2 & x_5 & x_5y_5 & y_5 & y_5^2 \end{pmatrix} \tag{63}$$

(a) V_5 の階数が 5 の場合

このとき任意の $\hat{S}$ に対して

$$\begin{pmatrix} a \\ b \\ c \\ d \\ e \end{pmatrix} \triangleq V_5^{-1} \begin{pmatrix} z_1 - \hat{S} \\ z_2 - \hat{S} \\ z_3 - \hat{S} \\ z_4 - \hat{S} \\ z_5 - \hat{S} \end{pmatrix} \tag{64}$$

のように a，b，c，d，e を定めると多項式

$$f'(x, y) = ax^2 + bx + cxy + dy + ey^2 + \hat{S} \tag{65}$$

はシェア v_1，v_2，v_3，v_4，v_5 の値と矛盾しない．さらにこれがシェア v_6 の値とも矛盾しないことが以下のように示される.

$$
\begin{aligned}
&f'(x_6, y_6) \\
&= \begin{pmatrix} x_6^2 & x_6 & x_6y_6 & y_6 & y_6^2 \end{pmatrix} \begin{pmatrix} a \\ b \\ c \\ d \\ e \end{pmatrix} + \hat{S} \\
&= \begin{pmatrix} \alpha_1 & \alpha_2 & \alpha_3 & \alpha_4 & \alpha_5 \end{pmatrix} V_5 V_5^{-1} \begin{pmatrix} z_1 - \hat{S} \\ z_2 - \hat{S} \\ z_3 - \hat{S} \\ z_4 - \hat{S} \\ z_5 - \hat{S} \end{pmatrix} + \hat{S} \\
&= \begin{pmatrix} \alpha_1 & \alpha_2 & \alpha_3 & \alpha_4 & \alpha_5 \end{pmatrix} \begin{pmatrix} z_1 \\ z_2 \\ z_3 \\ z_4 \\ z_5 \end{pmatrix} \\
&= z_6
\end{aligned} \tag{66}
$$

以上より，行列 V_5 で与えた a，b，c，d，e によって決まる多項式 $f'(x, y)$ は v_6 と矛盾せず成り立つことが言えた．よってこのとき，ある 5 個のシェアに任意の $\hat{S}$ を与えると，その $\hat{S}$ を満たす多項式が一意に決定し，かつその多項式は残った 1 個のシェアについても矛盾しないことが言えた．

よって，V_5 の階数が 5 のとき，定数列が他の 5 列に線形従属なら秘密情報 S に関する情報は得られない．

(b) V_5 の階数が 4 の場合

このとき行列 V_5 の部分行列を考えると，その階数が 4 となるものが存在する．一般性を損なうことなく次に示す行列 V_4 の階数は 4 とする．

$$
V_4 = \begin{pmatrix} x_1 & x_1y_1 & y_1 & y_1^2 \\ x_2 & x_2y_2 & y_2 & y_2^2 \\ x_3 & x_3y_3 & y_3 & y_3^2 \\ x_4 & x_4y_4 & y_4 & y_4^2 \end{pmatrix} \tag{67}
$$

ここで任意の $\hat{S}$，a に対して

$$
\begin{pmatrix} b \\ c \\ d \\ e \end{pmatrix} \triangleq V_4^{-1} \begin{pmatrix} z_1 - \hat{S} - ax_1^2 \\ z_2 - \hat{S} - ax_2^2 \\ z_3 - \hat{S} - ax_3^2 \\ z_4 - \hat{S} - ax_4^2 \end{pmatrix} \tag{68}
$$

のように b，c，d，e を定めると，多項式

$$f'(x,y) = ax^2 + bx + cxy + dy + ey^2 + \hat{S} \tag{69}$$

はシェア v_1，v_2，v_3，v_4 の値と矛盾しない．さらにこれがシェア v_5，v_6 の値とも矛盾しないことが以下のように示される．

まず $f'(x_5, y_5)$ と z_5 が等しいことを示す．行列 V_5 の階数が 4 であることから，ある β_1，β_2，β_3，β_4 が存在し，次式を満たす．

$$\begin{aligned}
&\begin{pmatrix} x_5^2 & x_5 & x_5y_5 & y_5 & y_5^2 \end{pmatrix} \\
&\quad = \begin{pmatrix} \beta_1 & \beta_2 & \beta_3 & \beta_4 \end{pmatrix} \begin{pmatrix} x_1^2 & x_1 & x_1y_1 & y_1 & y_1^2 \\ x_2^2 & x_2 & x_2y_2 & y_2 & y_2^2 \\ x_3^2 & x_3 & x_3y_3 & y_3 & y_3^2 \\ x_4^2 & x_4 & x_4y_4 & y_4 & y_4^2 \end{pmatrix}
\end{aligned} \tag{70}$$

$$\begin{pmatrix} z_5 - \hat{S} \end{pmatrix} = \begin{pmatrix} \beta_1 & \beta_2 & \beta_3 & \beta_4 \end{pmatrix} \begin{pmatrix} z_1 - \hat{S} \\ z_2 - \hat{S} \\ z_3 - \hat{S} \\ z_4 - \hat{S} \end{pmatrix} \tag{71}$$

ここで式 (70)，式 (71) を用いて次式より $f'(x_5, y_5)$ を計算する．

$$\begin{aligned}
&\begin{pmatrix} x_5 & x_5y_5 & y_5 & y_5^2 \end{pmatrix} \begin{pmatrix} b \\ c \\ d \\ e \end{pmatrix} + \hat{S} + ax_5^2 \\
&= \begin{pmatrix} \beta_1 & \beta_2 & \beta_3 & \beta_4 \end{pmatrix} V_4 V_4^{-1} \begin{pmatrix} z_1 - \hat{S} - ax_1 \\ z_2 - \hat{S} - ax_2 \\ z_3 - \hat{S} - ax_3 \\ z_4 - \hat{S} - ax_4 \end{pmatrix} \\
&\quad + \hat{S} + ax_5^2 \\
&= \begin{pmatrix} \beta_1 & \beta_2 & \beta_3 & \beta_4 \end{pmatrix} \begin{pmatrix} z_1 - \hat{S} - ax_1 \\ z_2 - \hat{S} - ax_2 \\ z_3 - \hat{S} - ax_3 \\ z_4 - \hat{S} - ax_4 \end{pmatrix} + \hat{S} + ax_5^2 \\
&= z_5 - \hat{S} - ax_5^2 + \hat{S} + ax_5^2 \\
&= z_5
\end{aligned} \tag{72}$$

以上より多項式 $f'(x,y)$ は 5 個目のシェアに矛盾しないことが言えた．

次に，式 (60)，式 (61) を用いて $f'(x_6, y_6)$ と z_6 が等しいことを示す．

$$
\begin{aligned}
&\begin{pmatrix} x_6 & x_6y_6 & y_6 & y_6^2 \end{pmatrix} \begin{pmatrix} b \\ c \\ d \\ e \end{pmatrix} + \hat{S} + ax_6^2 \\
&= \begin{pmatrix} \alpha_1 & \alpha_2 & \alpha_3 & \alpha_4 & \alpha_5 \end{pmatrix} \begin{pmatrix} x_1 & x_1y_1 & y_1 & y_1^2 \\ x_2 & x_2y_2 & y_2 & y_2^2 \\ x_3 & x_3y_3 & y_3 & y_3^2 \\ x_4 & x_4y_4 & y_4 & y_4^2 \\ x_5 & x_5y_5 & y_5 & y_5^2 \end{pmatrix} \begin{pmatrix} b \\ c \\ d \\ e \end{pmatrix} \\
&\quad + \hat{S} + ax_6^2 \\
&= \begin{pmatrix} \alpha_1 & \alpha_2 & \alpha_3 & \alpha_4 & \alpha_5 \end{pmatrix} \begin{pmatrix} z_1 - \hat{S} - ax_1^2 \\ z_2 - \hat{S} - ax_2^2 \\ z_3 - \hat{S} - ax_3^2 \\ z_4 - \hat{S} - ax_4^2 \\ z_5 - \hat{S} - ax_5^2 \end{pmatrix} \\
&\quad + \hat{S} + ax_6^2 \\
&= z_6 - \hat{S} - ax_6^2 + \hat{S} + ax_6^2 \\
&= z_6
\end{aligned} \tag{73}
$$

以上より多項式 $f'(x, y)$ は 6 個目のシェアに矛盾しないことが言えた．

よってこのとき，4 個のシェアについて秘密情報 $\hat{S}$ と多項式の 1 係数を任意に定めると，それを満たす多項式が一意に決定され，またその多項式は 5 個目のシェアと 6 個目のシェアに矛盾しないことが言えた．

これはどのような $\hat{S}$ を秘密情報と仮定しても，それらの仮定は同様に確からしいので，この場合，秘密情報 S に関する情報は得られない．

2. V_6 の階数が 4 の場合

一般性を損なうことなく，V_6 にて 5 行目と 6 行目が他の 4 行に線形従属とする．このときある α_1，α_2，α_3，α_4，β_1，β_2，β_3，β_4 が存在し，式 (74)，式 (75)，式 (76) を満たす．

$$
\begin{aligned}
&\begin{pmatrix} x_5^2 & x_5 & x_5y_5 & y_5 & y_5^2 & 1 \end{pmatrix} \\
&= \begin{pmatrix} \alpha_1 & \alpha_2 & \alpha_3 & \alpha_4 \end{pmatrix} V_4'
\end{aligned} \tag{74}
$$

$$
\begin{aligned}
&\begin{pmatrix} x_6^2 & x_6 & x_6y_6 & y_6 & y_6^2 & 1 \end{pmatrix} \\
&= \begin{pmatrix} \beta_1 & \beta_2 & \beta_3 & \beta_4 \end{pmatrix} V_4'
\end{aligned} \tag{75}
$$

$$\begin{pmatrix} z_5 \\ z_6 \end{pmatrix} = \begin{pmatrix} \alpha_1 & \alpha_2 & \alpha_3 & \alpha_4 \\ \beta_1 & \beta_2 & \beta_3 & \beta_4 \end{pmatrix} \begin{pmatrix} z_1 \\ z_2 \\ z_3 \\ z_4 \end{pmatrix} \tag{76}$$

ただし

$$V_4' = \begin{pmatrix} x_1^2 & x_1 & x_1y_1 & y_1 & y_1^2 & 1 \\ x_2^2 & x_2 & x_2y_2 & y_2 & y_2^2 & 1 \\ x_3^2 & x_3 & x_3y_3 & y_3 & y_3^2 & 1 \\ x_4^2 & x_4 & x_4y_4 & y_4 & y_4^2 & 1 \end{pmatrix} \tag{77}$$

とした.

ここで次のような行列 V_5 を考え，その階数により場合分けをする.

$$V_5 = \begin{pmatrix} x_1^2 & x_1 & x_1y_1 & y_1 & y_1^2 \\ x_2^2 & x_2 & x_2y_2 & y_2 & y_2^2 \\ x_3^2 & x_3 & x_3y_3 & y_3 & y_3^2 \\ x_4^2 & x_4 & x_4y_4 & y_4 & y_4^2 \\ x_5^2 & x_5 & x_5y_5 & y_5 & y_5^2 \end{pmatrix} \tag{78}$$

(a) V_5 の階数が 4 の場合

このとき行列 V_5 の部分行列を考えると，その階数が 4 となるものが存在する．一般性を損なうことなく，次に示す行列 V_4 の階数は 4 とする.

$$V_4 = \begin{pmatrix} x_1 & x_1y_1 & y_1 & y_1^2 \\ x_2 & x_2y_2 & y_2 & y_2^2 \\ x_3 & x_3y_3 & y_3 & y_3^2 \\ x_4 & x_4y_4 & y_4 & y_4^2 \end{pmatrix} \tag{79}$$

ここで任意の $\hat{S}$，a に対して

$$\begin{pmatrix} b \\ c \\ d \\ e \end{pmatrix} \triangleq V_4^{-1} \begin{pmatrix} z_1 - \hat{S} - ax_1^2 \\ z_2 - \hat{S} - ax_2^2 \\ z_3 - \hat{S} - ax_3^2 \\ z_4 - \hat{S} - ax_4^2 \end{pmatrix} \tag{80}$$

のように b，c，d，e を定めると多項式

$$f'(x, y) = ax^2 + bx + cxy + dy + ey^2 + \hat{S} \tag{81}$$

はシェア v_1，v_2，v_3，v_4 の値と矛盾しない．さらにこれがシェア v_5，v_6 の値とも矛盾しない

ことが以下のように示される．

$$
\begin{aligned}
&\begin{pmatrix} f'(x_5, y_5) \\ f'(x_6, y_6) \end{pmatrix} \\
&= \begin{pmatrix} x_5 & x_5y_5 & y_5 & y_5^2 \\ x_6 & x_6y_6 & y_6 & y_6^2 \end{pmatrix} \begin{pmatrix} b \\ c \\ d \\ e \end{pmatrix} + \begin{pmatrix} ax_5^2 + \hat{S} \\ ax_6^2 + \hat{S} \end{pmatrix} \\
&= \begin{pmatrix} \alpha_1 & \alpha_2 & \alpha_3 & \alpha_4 \\ \beta_1 & \beta_2 & \beta_3 & \beta_4 \end{pmatrix} V_4 V_4^{-1} \begin{pmatrix} z_1 - \hat{S} - ax_1^2 \\ z_2 - \hat{S} - ax_2^2 \\ z_3 - \hat{S} - ax_3^2 \\ z_4 - \hat{S} - ax_4^2 \end{pmatrix} \\
&\quad + \begin{pmatrix} ax_5^2 + \hat{S} \\ ax_6^2 + \hat{S} \end{pmatrix} \\
&= \begin{pmatrix} \alpha_1 & \alpha_2 & \alpha_3 & \alpha_4 \\ \beta_1 & \beta_2 & \beta_3 & \beta_4 \end{pmatrix} \begin{pmatrix} z_1 - \hat{S} - ax_1^2 \\ z_2 - \hat{S} - ax_2^2 \\ z_3 - \hat{S} - ax_3^2 \\ z_4 - \hat{S} - ax_4^2 \end{pmatrix} \\
&\quad + \begin{pmatrix} ax_5^2 + \hat{S} \\ ax_6^2 + \hat{S} \end{pmatrix} \\
&= \begin{pmatrix} z_5 - \hat{S} - ax_5^2 \\ z_6 - \hat{S} - ax_6^2 \end{pmatrix} + \begin{pmatrix} ax_5^2 + \hat{S} \\ ax_6^2 + \hat{S} \end{pmatrix} \\
&= \begin{pmatrix} z_5 \\ z_6 \end{pmatrix}
\end{aligned} \tag{82}
$$

以上より，式 (80) で定義した多項式 $f'(x, y)$ は v_5，v_6 に矛盾せず成り立つことが言えた．

よってこのとき $\hat{S}$ と多項式の 1 係数を任意の値に定めると，ある 4 個のシェアから多項式が一意に決定され，またその多項式は残りの 2 個のシェアについても矛盾せず成立する．秘密情報を任意の $\hat{S}$ と仮定しても，それらの仮定は同様に確からしいので，秘密情報 S に関する情報は得られない．

(b) V_5 の階数が 3 の場合

このとき行列 V_5 の部分行列のうち，階数が 3 となるものが存在する．一般性を損なうことなく，次に示す行列 V_3 の階数は 3 とする．

$$
V_3 = \begin{pmatrix} x_1y_1 & y_1 & y_1^2 \\ x_2y_2 & y_2 & y_2^2 \\ x_3y_3 & y_3 & y_3^2 \end{pmatrix} \tag{83}
$$

任意の $\hat{S}$，a，b に対して

$$\begin{pmatrix} c \\ d \\ e \end{pmatrix} \triangleq V_3^{-1} \begin{pmatrix} z_1 - \hat{S} - ax_1^2 - bx_1 \\ z_2 - \hat{S} - ax_2^2 - bx_2 \\ z_3 - \hat{S} - ax_3^2 - bx_3 \end{pmatrix} \tag{84}$$

のように c，d，e を定めると，多項式

$$f'(x, y) = ax^2 + bx + cxy + dy + ey^2 + \hat{S} \tag{85}$$

はシェア v_1，v_2，v_3 の値に矛盾しない．さらに式 (74)，式 (75)，式 (76)，式 (84) を用いると，これがシェア v_4，v_5，v_6 の値とも矛盾しないことが以下のように示される．

まず，行列 V_5 の 4 行目の行ベクトルが上の 3 行の行ベクトルに線形従属であることから v_4 について考える．ある γ_1，γ_2，γ_3 が存在し，それらは次式を満たす．

$$\begin{aligned} &\begin{pmatrix} x_4^2 & x_4 & x_4y_4 & y_4 & y_4^2 \end{pmatrix} \\ &= \begin{pmatrix} \gamma_1 & \gamma_2 & \gamma_3 \end{pmatrix} V_3' \end{aligned} \tag{86}$$

$$z_4 - \hat{S} = \begin{pmatrix} \gamma_1 & \gamma_2 & \gamma_3 \end{pmatrix} \begin{pmatrix} z_1 - \hat{S} \\ z_2 - \hat{S} \\ z_3 - \hat{S} \end{pmatrix} \tag{87}$$

ただし

$$V_3' = \begin{pmatrix} x_1^2 & x_1 & x_1y_1 & y_1 & y_1^2 \\ x_2^2 & x_2 & x_2y_2 & y_2 & y_2^2 \\ x_3^2 & x_3 & x_3y_3 & y_3 & y_3^2 \end{pmatrix} \tag{88}$$

とした．ここで $f'(x_4, y_4)$ について考える．

$$\begin{aligned} &f'(x_4, y_4) \\ &= \begin{pmatrix} x_4y_4 & y_4 & y_4^2 \end{pmatrix} \begin{pmatrix} c \\ d \\ e \end{pmatrix} + \left(\hat{S} + ax_4^2 + bx_4\right) \\ &= \begin{pmatrix} \gamma_1 & \gamma_2 & \gamma_3 \end{pmatrix} V_3 V_3^{-1} \begin{pmatrix} z_1 - \hat{S} - ax_1^2 - bx_1 \\ z_2 - \hat{S} - ax_2^2 - bx_2 \\ z_3 - \hat{S} - ax_3^2 - bx_3 \end{pmatrix} \\ &\quad + \left(\hat{S} + ax_4^2 + bx_4\right) \\ &= \begin{pmatrix} \gamma_1 & \gamma_2 & \gamma_3 \end{pmatrix} \begin{pmatrix} z_1 - \hat{S} - ax_1^2 - bx_1 \\ z_2 - \hat{S} - ax_2^2 - bx_2 \\ z_3 - \hat{S} - ax_3^2 - bx_3 \end{pmatrix} \\ &\quad + \left(\hat{S} + ax_4^2 + bx_4\right) \\ &= z_4 - \hat{S} - ax_4^2 - bx_4 + \hat{S} + ax_4^2 + bx_4 \\ &= z_4 \end{aligned} \tag{89}$$

以上より，v_4 は多項式 $f'(x,y)$ に矛盾せず成り立つことが言えた．

次に $f'(x_5,y_5)$，$f'(x_6,y_6)$ について考える．

$$\begin{aligned}
&\begin{pmatrix} f'(x_5,y_5) \\ f'(x_6,y_6) \end{pmatrix} \\
&= \begin{pmatrix} x_5y_5 & y_5 & y_5^2 \\ x_6y_6 & y_6 & y_6^2 \end{pmatrix} \begin{pmatrix} c \\ d \\ e \end{pmatrix} + \begin{pmatrix} \hat{S} + ax_5^2 + bx_5 \\ \hat{S} + ax_6^2 + bx_6 \end{pmatrix} \\
&= \begin{pmatrix} z_5 \\ z_6 \end{pmatrix}
\end{aligned} \tag{90}$$

以上より多項式 $f'(x,y)$ は v_4，v_5，v_6 に矛盾せず成り立つことが言えた．

よってこのとき $\hat{S}$ と多項式の 2 係数を任意の値に定めると，ある 3 個のシェアから多項式が一意に決定され，またその多項式は残りの 3 個のシェアについても矛盾せず成立する．秘密情報を任意の S と仮定しても，それらの仮定は同様に確からしいので，秘密情報 S に関する情報は得られない．

3. V_6 の階数が 3 の場合

一般性を損なうことなく，V_6 にて 4 行目と 5 行目と 6 行目が他の 3 行に線形従属とする．このときある α_1，α_2，α_3，β_1，β_2，β_3，γ_1，γ_2，γ_3 が存在し，次式を満たす．

$$\begin{aligned}
&\begin{pmatrix} x_4^2 & x_4 & x_4y_4 & y_4 & y_4^2 & 1 \end{pmatrix} \\
&= \begin{pmatrix} \alpha_1 & \alpha_2 & \alpha_3 \end{pmatrix} V_3' \qquad (91) \\
&\begin{pmatrix} x_5^2 & x_5 & x_5y_5 & y_5 & y_5^2 & 1 \end{pmatrix} \\
&= \begin{pmatrix} \beta_1 & \beta_2 & \beta_3 \end{pmatrix} V_3' \qquad (92) \\
&\begin{pmatrix} x_6^2 & x_6 & x_6y_6 & y_6 & y_6^2 & 1 \end{pmatrix} \\
&= \begin{pmatrix} \gamma_1 & \gamma_2 & \gamma_3 \end{pmatrix} V_3' \qquad (93) \\
&\begin{pmatrix} z_4 \\ z_5 \\ z_6 \end{pmatrix} = \begin{pmatrix} \alpha_1 & \alpha_2 & \alpha_3 \\ \beta_1 & \beta_2 & \beta_3 \\ \gamma_1 & \gamma_2 & \gamma_3 \end{pmatrix} \begin{pmatrix} z_1 \\ z_2 \\ z_3 \end{pmatrix} \qquad (94)
\end{aligned}$$

ただし

$$V_3' = \begin{pmatrix} x_1^2 & x_1 & x_1y_1 & y_1 & y_1^2 & 1 \\ x_2^2 & x_2 & x_2y_2 & y_2 & y_2^2 & 1 \\ x_3^2 & x_3 & x_3y_3 & y_3 & y_3^2 & 1 \end{pmatrix} \tag{95}$$

とした．次のような行列 V_5 を考え，その階数により場合分けをする．

$$V_5 = \begin{pmatrix} x_1^2 & x_1 & x_1y_1 & y_1 & y_1^2 \\ x_2^2 & x_2 & x_2y_2 & y_2 & y_2^2 \\ x_3^2 & x_3 & x_3y_3 & y_3 & y_3^2 \\ x_4^2 & x_4 & x_4y_4 & y_4 & y_4^2 \\ x_5^2 & x_5 & x_5y_5 & y_5 & y_5^2 \end{pmatrix} \tag{96}$$

(a) V_5 の階数が 3 の場合

このとき行列 V_5 の部分行列を考えると，その階数が 3 となるものが存在する．一般性を損なうことなく，次式に示す行列 V_3 の階数は 3 とする．

$$V_3 = \begin{pmatrix} x_1y_1 & y_1 & y_1^2 \\ x_2y_2 & y_2 & y_2^2 \\ x_3y_3 & y_3 & y_3^2 \end{pmatrix} \tag{97}$$

ここで任意の $\hat{S}$，a，b に対して

$$\begin{pmatrix} c \\ d \\ e \end{pmatrix} \triangleq V_3^{-1} \begin{pmatrix} z_1 - \hat{S} - ax_1^2 - bx_1 \\ z_2 - \hat{S} - ax_2^2 - bx_2 \\ z_3 - \hat{S} - ax_3^2 - bx_3 \end{pmatrix} \tag{98}$$

のように c，d，e を定めると，多項式

$$f'(x, y) = ax^2 + bx + cxy + dy + ey^2 + \hat{S} \tag{99}$$

はシェア v_1，v_2，v_3 の値と矛盾しない．さらにこれがシェア v_4，v_5，v_6 の値とも矛盾しないことが以下のように示される．

式 (98) を用いて $f'(x_4, y_4)$，$f'(x_5, y_5)$，$f'(x_6, y_6)$ を計算する．

$$
\begin{aligned}
&\begin{pmatrix} f'(x_4, y_4) \\ f'(x_5, y_5) \\ f'(x_6, y_6) \end{pmatrix} \\
&= \begin{pmatrix} x_4 y_4 & y_4 & y_4^2 \\ x_5 y_5 & y_5 & y_5^2 \\ x_6 y_6 & y_6 & y_6^2 \end{pmatrix} \begin{pmatrix} c \\ d \\ e \end{pmatrix} + \begin{pmatrix} ax_4^2 + bx_4 + \hat{S} \\ ax_5^2 + bx_5 + \hat{S} \\ ax_6^2 + bx_6 + \hat{S} \end{pmatrix} \\
&= \begin{pmatrix} \alpha_1 & \alpha_2 & \alpha_3 \\ \beta_1 & \beta_2 & \beta_3 \\ \gamma_1 & \gamma_2 & \gamma_3 \end{pmatrix} V_3 V_3^{-1} \begin{pmatrix} z_1 - \hat{S} - ax_1^2 - bx_1 \\ z_2 - \hat{S} - ax_2^2 - bx_2 \\ z_3 - \hat{S} - ax_3^2 - bx_3 \end{pmatrix} \\
&\quad + \begin{pmatrix} ax_4^2 + bx_4 + \hat{S} \\ ax_5^2 + bx_5 + \hat{S} \\ ax_6^2 + bx_6 + \hat{S} \end{pmatrix} \\
&= \begin{pmatrix} \alpha_1 & \alpha_2 & \alpha_3 \\ \beta_1 & \beta_2 & \beta_3 \\ \gamma_1 & \gamma_2 & \gamma_3 \end{pmatrix} \begin{pmatrix} z_1 - \hat{S} - ax_1^2 - bx_1 \\ z_2 - \hat{S} - ax_2^2 - bx_2 \\ z_3 - \hat{S} - ax_3^2 - bx_3 \end{pmatrix} \\
&\quad + \begin{pmatrix} ax_4^2 + bx_4 + \hat{S} \\ ax_5^2 + bx_5 + \hat{S} \\ ax_6^2 + bx_6 + \hat{S} \end{pmatrix} \\
&= \begin{pmatrix} z_4 - \hat{S} - ax_4^2 - bx_4 \\ z_5 - \hat{S} - ax_5^2 - bx_5 \\ z_6 - \hat{S} - ax_6^2 - bx_6 \end{pmatrix} + \begin{pmatrix} ax_4^2 + bx_4 + \hat{S} \\ ax_5^2 + bx_5 + \hat{S} \\ ax_6^2 + bx_6 + \hat{S} \end{pmatrix} \\
&= \begin{pmatrix} z_4 \\ z_5 \\ z_6 \end{pmatrix}
\end{aligned} \tag{100}
$$

以上より多項式 $f'(x, y)$ は v_4，v_5，v_6 に矛盾せず成り立つことが言えた．

よってこのとき $\hat{S}$ と多項式の 2 係数を任意の値に定めると，ある 3 個のシェアから多項式が一意に決定され，またその多項式は残りの 3 個のシェアについても矛盾せず成立する．秘密情報を任意の $\hat{S}$ と仮定しても，それらの仮定は同様に確からしいので，秘密情報 S に関する情報は得られない．

(b) V_5 の階数が 2 の場合

シェアが 5 個集まった場合の行列 V_5 の階数が 2 の場合に帰着し，このときシェア v_1，v_2，v_3，v_4，v_5 のうちどの 3 個を選んでもアクセス集合となるので，この場合は除外して考えて良い．

4. V_6 の階数が 2 以下の場合

このとき変数列から任意の 5 行を選び 5 行 5 列の部分行列を考えると，その階数は高々 2 である．これはシェアが 5 個与えられたときの，行列 V_5 の階数が 2 の場合に帰着し，集まった

6 個のシェアのうちどの 3 個を選んでもアクセス集合となることが言える．よって今この場合は除外して考えて良い．

以上より，アクセス集合を部分集合に含まない 6 個のシェアが集まったときにアクセス集合となる条件は，行列 V_6 の階数が 6 となることである．□

4　アクセス集合の例

上記の定理を基とすれば，各ユーザーが管理するシェアを 1 個としたまま実現出来る一般型アクセス構造を示すことが出来る．その一例として，前書きで提示した例の実現法について述べる．有限体は 7 を法とする整数とする．

今 6 個のシェアがあり，それらの位置は $v_1 = (1, 2)$，$v_2 = (2, 4)$，$v_3 = (3, 6)$，$v_4 = (5, 4)$，$v_5 = (3, 5)$，$v_6 = (1, 6)$ とする．ここで $\{v_1, v_2, v_3\}$ について

$$V_3 = \begin{pmatrix} x_1 & y_1 \\ x_2 & y_2 \\ x_3 & y_3 \end{pmatrix} = \begin{pmatrix} 1 & 2 \\ 2 & 4 \\ 3 & 6 \end{pmatrix} \tag{101}$$

とおいて定理 1 を考えると，上に挙げた 3 個はアクセス集合であることが分かる．同様に定理 3 を考えると，詳細は省略するが $\{v_1, v_2, v_4, v_5, v_6\}$，$\{v_1, v_3, v_4, v_5, v_6\}$，$\{v_2, v_3, v_4, v_5, v_6\}$ もアクセス集合であることが分かる．これらを部分集合に持つ組み合わせもアクセス集合であり，逆に定理の必要性よりその他の組み合わせは非アクセス集合である．

5　アルゴリズムの実装

上で述べた定理を基に，7 個のシェアを入力したとき，アクセス集合を出力するプログラムを作成した．図 1 にそのアルゴリズムを示す．またソースコードを付録に示す．

1. シェアを読み込む.
2. 3 個の組み合わせを選び，次へ進む．もし全ての組み合わせを終えたら 5 へ進む.
3. 定理 1 を用いて判定を行う．アクセス集合なら次へ進み，偽なら 2 へ戻る.
4. 判定に用いた 3 個のシェアの位置を表示し，メモリに組み合わせを保存して 2 に戻る.
5. 5 個の組み合わせを選ぶ．メモリに保存された組み合わせを含まなければ次へ進み，もし全ての組み合わせを終えたら 8 へ進む.
6. 定理 3 を用いて判定を行う．アクセス集合なら次へ進み，偽なら 5 へ戻る.
7. 判定に用いた 5 個のシェアの位置を表示し，メモリに組み合わせを保存して 5 に戻る.
8. 6 個の組み合わせを選ぶ．メモリに保存された組み合わせを含まなければ次へ進み，もし全ての組み合わせを終えたら 11 へ進む.
9. 定理 4 を用いて判定を行う．アクセス集合なら次へ進み，偽なら 8 へ戻る.
10. 判定に用いた 6 個のシェアの位置を表示し 8 に戻る.
11. 終了

図 1　シェアの位置を入力したときアクセス集合を出力するアルゴリズム

6　まとめと今後の課題

秘密分散の手法として，2 変数 2 次の多項式補間法を提案した．この方法を用いるとシェアの個数や x–y 平面上での位置により復元の可否が決まり，全てのユーザーが管理するシェアを 1 個とした状態で一般型アクセス構造が実現出来る場合がある．本論文では，シェアの集合がアクセス集合であるための位置に関する必要十分条件を代数的に明らかにした．本論文の定理に従えば，シェアの位置を入力したときにアクセス構造を出力するアルゴリズムを作成することが出来る.

本論文の定理を基に全探索を行えば，所望のアクセス構造が可能かどうか，またどの位置のシェアを生成すれば良いか分かるが，今後はこれを効率良く行う方法について考えていきたい.

また次数や変数の数を増やすとアクセス構造はより広がると思われるので，それらについて考えることも興味深い.

謝辞

この研究を進めるにあたり指導していただいた西新幹彦准教授，また文献 [5] を教えてくださった長岡技術科学大学の武井由智准教授に感謝する.

参考文献

[1] A.Shamir, “How to share a secret,” Comm. ACM, vol.22, no.11, pp.612-613, 1979.

[2] 岩本貢，山本博資，“一般型アクセス構造に対する強い秘密保護特性をもつランプ型秘密分散法”，第 27 回情報理論とその応用シンポジウム予稿集，pp.331–334，2004.

[3] 山本博資，“秘密分散法とそのバリエーション”，数理解析研究所講究録，1361 巻，pp.19–31，2004.

[4] 柏倉英明，西新幹彦，“2 変数の多項式補間法による秘密分散法のアクセス構造について”，IEICE Technical Report, vol.109, No.444, pp.445–448, Mar. 2010.

[5] T.Tassa, N.Dyn, “Multipartite Secret Sharing by Bivariate Interpolation,” J.Cryptol, vol.22, no.2, pp.227-258, 2009.

[6] 今井秀樹，符号理論，電子情報通信学会，2002.

[7] T.Matsumoto，H.Imai, “Public Quadratic polynomial-tuples for efficient signature-verification and message-encryption,” EUROCRYPT’88, Springer-Verlog 1988, pp.419–453.

[8] M.Naor, B.Pinkas, O.Reingold, “Distributed Pseudo-Random Functions and KDCs,” Advances in Cryptology-Eurocrypt ’99, LNCS 1592, Springer-Verlag, pp.327-346, 1999.

[9] 伊藤充，斉藤明，西関隆夫，“一般型アクセス構造を実現する秘密共有法”，電子情報通信学会論文誌 A，vol.J71-A，No.8，pp.1592–1598，Aug. 1988.

[10] 佐武一郎，行列と行列式，裳華房，1958.

付録 A　有限体のサイズについて

定理 1 の証明の脚注にて，2 変数 2 次の多項式補間法では有限体 $\mathbb{F}$ のサイズは 7 以上であれば十分であることが記されている．ここではその証明を示す．

（証明）定理の証明内で有限体のサイズが問題になるのは，必要性を示す際に新たなシェアを追加するときである．よって，それぞれの場合において位数が 7 以上であれば常に新たなシェアが追加出来ることを示す．

1. 定理 1 について

定理 1 の証明では，一般性を損なうことなく $x_1 \neq 0$，かつある α に対して

$$y_1 = \alpha x_1, \quad y_2 \neq \alpha x_2, \quad y_3 \neq \alpha x_3 \tag{4}$$

が成り立つと仮定し，

$$(x_4, y_4) \neq (x_1, y_1) \tag{5}$$

$$y_4 = \alpha x_4 \tag{6}$$

$$y_5 \neq \alpha x_5 \tag{7}$$

$$\begin{vmatrix} x_2 - x_5 & y_2 - y_5 \\ x_3 - x_5 & y_3 - y_5 \end{vmatrix} \neq 0 \tag{8}$$

を満たす位置にシェア v_4，v_5 を選んでいる．ここで式 (4)，式 (5)，式 (6)，式 (7)，式 (8) は再掲である．

まず v_4 について考える．式 (5)，式 (6) より，位数が 7 以上であればシェア v_4 は常に選べることは自明である．

次に v_5 について考える．まず位数が 7 の場合を考えると，シェアの位置の候補は，原点を除いて 48 箇所である．式 (7) を考えると，v_5 の位置の候補は 42 箇所になる．この 42 箇所の中に v_1，v_4 はない．また式 (8) の左辺を計算すると

$$\begin{vmatrix} x_2 - x_5 & y_2 - y_5 \\ x_3 - x_5 & y_3 - y_5 \end{vmatrix} = x_5(y_2 + y_3) + y_5(x_3 - x_2) + x_2 y_3 - x_3 y_2 \tag{102}$$

となる．これは x_5 と y_5 の一次式になっている．よって，式 (8) によって更に v_5 の位置の候補を狭めると，残りの候補は少なくとも 35 箇所になる．この 35 箇所の中に v_2，v_3 はない．よって位数が 7 であれば，定理 1 は成立する．

位数が 7 より大きくとも，上記と同様に示すことが出来る．以上より，位数が 7 以上であれば定理 1 は成立する．

2. 定理 2 について

定理 2 の証明ではシェアを追加している場所は 3 箇所ある．

まず 2 個からなる同値類がある場合について考える．このとき v_1 と v_2 が同値であるとし，

$$\begin{vmatrix} x_4 - x_3 & y_4 - y_3 \\ x_5 - x_3 & y_5 - y_3 \end{vmatrix} \neq 0 \tag{15}$$

$$x_1 y_5 - x_5 y_1 \neq 0 \tag{16}$$

を満たす位置にシェア v_5 を追加する．式 (15) は式 (8)，式 (16) は式 (7) と同じ形をしており，前述と同様にすれば，位数が 7 以上であればシェア v_5 は常に選べることが言える．

次に 2 個からなる同値類がない場合について考える．証明中では

$$\begin{vmatrix} x_1 - x_3 & y_1 - y_3 \\ x_2 - x_3 & y_2 - y_3 \end{vmatrix} \neq 0 \tag{23}$$

を満たす 3 個があるかどうかで場合分けをし，シェア v_5 を追加する．しかし，どちらの場合でも追加するシェア v_5 は

$$x_4 y_5 - x_5 y_4 = 0 \tag{103}$$

を満たす位置に選ばれている．仮定より今 2 個からなる同値類はないため，v_5 の位置の候補は 5 箇所ある．よって位数が 7 であれば定理 2 は成立する．

位数が 7 より大きくとも，上記と同様に示すことが出来る．以上より，位数が 7 以上であれば定理 2 は成立する．

定理 3 と定理 4 の証明内では新たにシェアを追加することがない．よって位数が 7 以上であれば，全ての定理が成立することが言えた．□

付録 B　ソースコード

B.1　後述の 2 つのソードコードで用る，GF(2^8) と行列のクラスのソースコード

```
#include <iostream>
#include <fstream>
using namespace std;

class GF256
{
private:
    unsigned char num;

public:
    GF256(): num(0) {}
    GF256(int n): num(n) {}
    GF256 operator+(const GF256 &b);
    GF256 operator-(const GF256 &b);
    GF256 operator*(const GF256 &b);
    GF256 operator/(const GF256 &b);
    GF256 operator=(const int b);
    operator int();
};
```

```
inline GF256 GF256::operator+(const GF256 &b)
{
    return(GF256(num ^ b.num));
}

inline GF256 GF256::operator-(const GF256 &b)
{
    return(GF256(num ^ b.num));
}

GF256 GF256::operator*(const GF256 &b)
{
    unsigned char a7b = (num & 0x80)? b.num: 0;
    unsigned char a6b = (num & 0x40)? b.num: 0;
    unsigned char a5b = (num & 0x20)? b.num: 0;
    unsigned char a4b = (num & 0x10)? b.num: 0;
    unsigned char a3b = (num & 0x08)? b.num: 0;
    unsigned char a2b = (num & 0x04)? b.num: 0;
    unsigned char a1b = (num & 0x02)? b.num: 0;
    unsigned char a0b = (num & 0x01)? b.num: 0;
    unsigned char y0 = (a0b)^(a1b<<1)^(a2b<<2)^(a3b<<3)^(a4b<<4)^(a5b<<5)^(a6b<<6)^(a7b<<7);
    unsigned char y1 = (a1b>>7)^(a2b>>6)^(a3b>>5)^(a4b>>4)^(a5b>>3)^(a6b>>2)^(a7b>>1);
    unsigned char p = y0;
    p ^= (y1 << 4) & 0xf0;
    p ^= (y1 << 3) & 0xf8;
    p ^= (y1 << 2) & 0xfc;
    p ^= (y1 >> 0) & 0x0f;
    p ^= (y1 >> 1) & 0x08;
    p ^= (y1 >> 2) & 0x14;
    p ^= (y1 >> 3) & 0x04;
    p ^= (y1 >> 4) & 0x03;
    p ^= (y1 >> 5) & 0x03;
    p ^= (y1 >> 6) & 0x01;
    return(GF256(p));
}

GF256 GF256::operator/(const GF256 &b)
{
    unsigned char c[8];
    unsigned char e[8];

    c[7] = (b.num << 0) ^ (b.num >> 4) ^ (b.num >> 5) ^ (b.num >> 6);
    c[6] = (b.num << 1) ^ (b.num >> 3) ^ (b.num >> 4) ^ (b.num >> 5);
    c[5] = (b.num << 2) ^ (b.num >> 2) ^ (b.num >> 3) ^ (b.num >> 4);
    c[4] = (b.num << 3) ^ (b.num >> 1) ^ (b.num >> 2) ^ (b.num >> 3) ^ (b.num >> 7);
    c[3] = (b.num << 4) ^ (b.num >> 1) ^ (b.num >> 2) ^ (b.num >> 4) ^ (b.num >> 5);
    c[2] = (b.num << 5) ^ (b.num >> 1) ^ (b.num >> 3) ^ (b.num >> 5) ^ (b.num >> 6);
    c[1] = (b.num << 6) ^ (b.num >> 2) ^ (b.num >> 6) ^ (b.num >> 7);
    c[0] = (b.num << 7) ^ (b.num >> 1) ^ (b.num >> 5) ^ (b.num >> 6) ^ (b.num >> 7);
    for (int i = 0; i < 8; i++){
        e[i] = !!(num & (1 << i));
    }

    for (int i = 0; i < 8; i++){
        int j;
        for (j = i; j < 8 && !(c[j] & (1 << i)); j++){
            ;
        }
        if (j >= 8){
            cout<<"divided by 0"<<endl;
            exit(1);
        }
        unsigned char tmp;
        tmp = c[i], c[i] = c[j], c[j] = tmp;
        tmp = e[i], e[i] = e[j], e[j] = tmp;
        for (j = 0; j < 8; j++){
            if (i != j && (c[j] & (1 << i))){
                c[j] ^= c[i];
```

```
                e[j] ^= e[i];
            }
        }
    }
    unsigned char div = 0;
    for (int i = 0; i < 8; i++){
        div |= (e[i])? 0x80 >> i: 0;
    }
    return(GF256(div));
}

inline GF256 GF256::operator=(const int b)
{
    num = (unsigned char)b;
    return(*this);
}

inline GF256::operator int()
{
    return((int)num);
}


template<typename TYPE>
class matrix
{
private:
    int size;
    TYPE *elem;

public:
    matrix();
    explicit matrix(int s){
        size = s;
        elem = new TYPE[s * s];
        return;
    }
    matrix(const matrix<TYPE> &s){
        size = s.size;
        elem = new TYPE[s.size * s.size];
        int elements = size * size;
        for (int k = 0; k < elements; k++){
            elem[k] = s.elem[k];
        }
        return;
    }
    ~matrix(){
        delete[] elem;
        return;
    }
    TYPE &e(int j, int i);
    matrix<TYPE> &unit(void);
    matrix<TYPE> &operator=(const matrix<TYPE> &b);
    matrix<TYPE> operator~(void);
    matrix<TYPE> operator*(const matrix<TYPE> &b);
    int rank(void);
    TYPE vector(int i, TYPE *v);
};

template<typename TYPE>
inline TYPE &matrix<TYPE>::e(int j, int i)
{
    return(elem[j * size + i]);
}

template<typename TYPE>
matrix<TYPE> &matrix<TYPE>::unit(void)
{
    for (int j = 0; j < size; j++){
```

```
        for (int i = 0; i < size; i++){
            elem[j * size + i] = (i == j)? TYPE(1): TYPE(0);
        }
    }
    return(*this);
}

template<typename TYPE>
matrix<TYPE> &matrix<TYPE>::operator=(const matrix<TYPE> &b)
{
    if (b.size == size){
        int elements = size * size;
        for (int k = 0; k < elements; k++){
            elem[k] = b.elem[k];
        }
    }
    return(*this);
}

template<typename TYPE>
matrix<TYPE> matrix<TYPE>::operator~()
{
    matrix<TYPE> a(size);
    matrix<TYPE> b(size);
    a = *this;
    b.unit();

    for (int k = 0; k < size; k++){
        TYPE t = a.e(k, k);
        for ( int m=k+1; (int)t==0 && m < size; m++){
            for( int i = 0; i < size; i++){
                a.e(k,i)=a.e(k,i)+a.e(m,i);
                b.e(k,i)=b.e(k,i)+b.e(m,i);
            }
            t = a.e(k, k);
        }

        if((int)t==0){
            cout<<"逆行列計算中に、何かを０で割りました"<<endl;
        }

        for (int i = 0; i < size; i++){
            a.e(k, i) = a.e(k, i) / t;
            b.e(k, i) = b.e(k, i) / t;
        }
        for (int j = 0; j < size; j++){
            if (j != k){
                t = a.e(j, k);
                for (int i = 0; i < size; i++){
                    a.e(j, i) = a.e(j, i) - a.e(k, i) * t;
                    b.e(j, i) = b.e(j, i) - b.e(k, i) * t;
                }
            }
        }
    }
    return(b);
}

template<typename TYPE>
matrix<TYPE> matrix<TYPE>::operator*(const matrix<TYPE> &b)
{
    matrix<TYPE> prod(size);
    for (int j = 0; j < size; j++){
        for (int i = 0; i < size; i++){
            TYPE e = elem[j * size] * b.elem[i];
            for (int k = 1; k < size; k++){
                e = e + elem[j * size + k] * b.elem[k * size + i];
            }
            prod.elem[j * size + i] = e;
```

```
        }
    }
    return(prod);
}

template<typename TYPE>
TYPE matrix<TYPE>::vector(int i, TYPE *v)
{
    TYPE ans;
    ans = TYPE(0);
    for (int j = 0; j < size; j++){
        ans = ans + elem[i * size + j] * v[j];
    }
    return(ans);
}

template<typename TYPE>
int matrix<TYPE>::rank(void)
{
    matrix<TYPE> x(size);
    x = *this;

    for(int i=0; i<size; i++){
        for(int m= i+1; m < size && (int)x.e(i,i)==0; m++){
            for(int n=0; n < size; n++){
                x.e(i,n)=x.e(i,n)+x.e(m,n);
            }
        }

        TYPE a = x.e(i,i);

        if((int)a!=0){
            for(int j=0; j<size; j++){
                x.e(i,j) = x.e(i,j)/a;
            }

            for(int k=0; k<size; k++){
                if(i!=k){
                    TYPE b = x.e(k,i);
                    for(int l=0; l<size; l++){
                        x.e(k,l)=x.e(k,l) - (b*x.e(i,l));
                    }
                }
            }
        }
    }

    int r = 0;
    int v = 0;
    for(int n=0; n<size; n++){
        for(int m=0; m<size;m++){
            int c = (int)x.e(n,m);
            if(c!=0){
                v = 1;
            }
        }
        if(v==1){
            r++;
        }
        if(v>1){
        cout<<"ランク計算に不具合があります"<<endl;
        }
        v=0;
    }

    if(r>size){
        cout<<"階数カウントにエラー発生"<<endl;
    }
    return(r);
```

```
}
```

B.2 シェアの位置を 7 回入力すると，それらのシェアをテキストファイルに出力するプログラム

```
#include <iostream>
#include <fstream>
#include "gfmatrix.h"
using namespace std;

int main()
{
    GF256 a,b,c,d,e,s;
    int p;
    a = 26;
    b = 5;
    c = 156;
    d = 200;
    e = 90;

    s = 64;
    GF256 x,y;
    int n=7;
    int k = 0;
    ofstream fileout;
    ifstream filein;
    cout<<n<<"個のシェアを用意してください"<<endl;

    fileout.open("share.txt");
    fileout<<n<<endl;
    do{

        cout<<k+1<<"回目の入力です"<<endl;
        cout<<"X の値を入力してください。"<<endl;
        cin>>p;
        x = GF256(p);
        cout<<"Y の値を入力してください。"<<endl;
        cin>>p;
        y = GF256(p);

        GF256 z = a*x*x + b*x + c*x*y + d*y + e*y*y + s;

        fileout<<(int)x<<" "<<(int)y<<" "<<(int)z<<endl;
        k++;
        cout<<"---------X="<<(int)x<<",Y="<<(int)y<<"---------"<<endl;
    }while(k<n);

    fileout.close();
    return 0;
}
```

B.3 シェアを入力すると，アクセス集合を出力するプログラムのソースコード

```
#include <iostream>
#include <fstream>
#include"gfmatrix.h"
using namespace std;

GF256 t3(GF256 x1, GF256 y1, GF256 z1, GF256 x2, GF256 y2, GF256 z2, GF256 x3, GF256 y3, GF256 z3)
{
    matrix<GF256> p(3);
```

```
    p.e(0,0)=x1;
    p.e(0,1)=y1;
    p.e(0,2)=0;

    p.e(1,0)=x2;
    p.e(1,1)=y2;
    p.e(1,2)=0;

    p.e(2,0)=x3;
    p.e(2,1)=y3;
    p.e(2,2)=0;

    cout<<"("<<x1<<","<<y1<<"),("<<x2<<","<<y2<<"),("<<x3<<","<<y3<<") はアクセス集合"<<endl;

    matrix<GF256> v(3);

    v.e(0,0)=x1*x1;
    v.e(0,1)=x1;
    v.e(0,2)=1;

    v.e(1,0)=x2*x2;
    v.e(1,1)=x2;
    v.e(1,2)=1;

    v.e(2,0)=x3*x3;
    v.e(2,1)=x3;
    v.e(2,2)=1;

    matrix<GF256> w(3);
    w=~v;

    GF256 *z = new GF256[3];
    z[0]=z1;
    z[1]=z2;
    z[2]=z3;

    GF256 S=w.vector(2,z);
    cout<<"S="<<S<<endl;
    delete z;

    return(0);
}

int line(GF256 x1, GF256 y1, GF256 x2, GF256 y2, GF256 x3, GF256 y3, GF256 x4, GF256 y4, GF256 x5, GF256 y5)
{
    matrix<GF256> p(5);

    p.e(0,0)=x1*x1;
    p.e(0,1)=x1;
    p.e(0,2)=x1*y1;
    p.e(0,3)=y1;
    p.e(0,4)=y1*y1;

    p.e(1,0)=x2*x2;
    p.e(1,1)=x2;
    p.e(1,2)=x2*y2;
    p.e(1,3)=y2;
    p.e(1,4)=y2*y2;

    p.e(2,0)=x3*x3;
    p.e(2,1)=x3;
    p.e(2,2)=x3*y3;
    p.e(2,3)=y3;
    p.e(2,4)=y3*y3;

    p.e(3,0)=x4*x4;
    p.e(3,1)=x4;
    p.e(3,2)=x4*y4;
    p.e(3,3)=y4;
```

```
    p.e(3,4)=y4*y4;

    p.e(4,0)=x5*x5;
    p.e(4,1)=x5;
    p.e(4,2)=x5*y5;
    p.e(4,3)=y5;
    p.e(4,4)=y5*y5;

    for(int i=0; i<5; i++){
        for(int m= i+1; m < 5 && (int)p.e(i,i)==0; m++){
            for(int n=0; n < 5 ;n++){
                p.e(i,n)=p.e(i,n)+p.e(m,n);
            }
        }

        GF256 a = p.e(i,i);

        if((int)a!=0){
            for(int j=0; j<5; j++){
                p.e(i,j) = p.e(i,j)/a;
            }

            for(int k=0; k<5; k++){
                if(i!=k){
                    GF256 b = p.e(k,i);
                    for(int l=0; l<5; l++){
                        p.e(k,l)=p.e(k,l) - (b*p.e(i,l));
                    }
                }
            }
        }
    }

    int line=0;
    for(int c=0; c<5;c++){
        if((int)p.e(c,c)==0)
            line = c;
    }

    return(line);
}

GF256 t5(GF256 x1, GF256 y1, GF256 z1, GF256 x2, GF256 y2, GF256 z2,GF256 x3, GF256 y3,GF256 z3,
GF256 x4, GF256 y4,GF256 z4, GF256 x5, GF256 y5,GF256 z5)
{
    matrix<GF256> p(5);

    p.e(0,0)=x1*x1;
    p.e(0,1)=x1;
    p.e(0,2)=x1*y1;
    p.e(0,3)=y1;
    p.e(0,4)=y1*y1;

    p.e(1,0)=x2*x2;
    p.e(1,1)=x2;
    p.e(1,2)=x2*y2;
    p.e(1,3)=y2;
    p.e(1,4)=y2*y2;

    p.e(2,0)=x3*x3;
    p.e(2,1)=x3;
    p.e(2,2)=x3*y3;
    p.e(2,3)=y3;
    p.e(2,4)=y3*y3;

    p.e(3,0)=x4*x4;
    p.e(3,1)=x4;
    p.e(3,2)=x4*y4;
    p.e(3,3)=y4;
```

```
    p.e(3,4)=y4*y4;

    p.e(4,0)=x5*x5;
    p.e(4,1)=x5;
    p.e(4,2)=x5*y5;
    p.e(4,3)=y5;
    p.e(4,4)=y5*y5;

    int l =line(x1,y1,x2,y2,x3,y3,x4,y4,x5,y5);

    for(int i=0;i<5;i++){
        p.e(i,l)=1;
    }

    if(p.rank()==5){
        cout<<"("<<x1<<","<<y1<<"),("<<x2<<","<<y2<<"),("<<x3<<","<<y3<<"),
        ("<<x4<<","<<y4<<"),("<<x5<<","<<y5<<") はアクセス集合"<<endl;
        matrix<GF256> v(5);
        v=~p;
        GF256 *z = new GF256[5];
        z[0]=z1;
        z[1]=z2;
        z[2]=z3;
        z[3]=z4;
        z[4]=z5;
        GF256 S=v.vector(l,z);
        cout<<"S="<<S<<endl;
        delete z;

        return(1);
    }else{
        return(0);
    }
}

GF256 t6(GF256 x1, GF256 y1, GF256 z1,GF256 x2, GF256 y2,GF256 z2, GF256 x3,
 GF256 y3,GF256 z3, GF256 x4, GF256 y4,GF256 z4, GF256 x5, GF256 y5,GF256 z5, GF256 x6 ,GF256 y6,GF256 z6)
{
    matrix<GF256> p(6);

    p.e(0,0)=x1*x1;
    p.e(0,1)=x1;
    p.e(0,2)=x1*y1;
    p.e(0,3)=y1;
    p.e(0,4)=y1*y1;
    p.e(0,5)=1;

    p.e(1,0)=x2*x2;
    p.e(1,1)=x2;
    p.e(1,2)=x2*y2;
    p.e(1,3)=y2;
    p.e(1,4)=y2*y2;
    p.e(1,5)=1;

    p.e(2,0)=x3*x3;
    p.e(2,1)=x3;
    p.e(2,2)=x3*y3;
    p.e(2,3)=y3;
    p.e(2,4)=y3*y3;
    p.e(2,5)=1;

    p.e(3,0)=x4*x4;
    p.e(3,1)=x4;
    p.e(3,2)=x4*y4;
    p.e(3,3)=y4;
    p.e(3,4)=y4*y4;
    p.e(3,5)=1;

    p.e(4,0)=x5*x5;
```

```
    p.e(4,1)=x5;
    p.e(4,2)=x5*y5;
    p.e(4,3)=y5;
    p.e(4,4)=y5*y5;
    p.e(4,5)=1;

    p.e(5,0)=x6*x6;
    p.e(5,1)=x6;
    p.e(5,2)=x6*y6;
    p.e(5,3)=y6;
    p.e(5,4)=y6*y6;
    p.e(5,5)=1;

    if(p.rank()==6){
        cout<<"("<<x1<<","<<y1<<"),("<<x2<<","<<y2<<"),("<<x3<<","<<y3<<"),("<<x4<<","<<y4<<"),
        ("<<x5<<","<<y5<<"),("<<x6<<","<<y6<<") はアクセス集合"<<endl;
        matrix<GF256> v(6);
        v=~p;
        GF256 *z = new GF256[6];
        z[0]=z1;
        z[1]=z2;
        z[2]=z3;
        z[3]=z4;
        z[4]=z5;
        z[5]=z6;
        GF256 S=v.vector(5,z);
        cout<<"S="<<S<<endl;
        delete z;
    }
    return(0);

}

int ch5(int *mem, int c , int i, int j, int k, int l,int m){
    int sn[5]={i,j,k,l,m};

    int ans=1;

    for(int mn=0;mn<c;mn+=3){
        for(int i=0;i<5;i++){
            if(sn[i]==mem[mn]){
                for(int j=i+1;j<5;j++){
                    if(sn[j]==mem[mn+1]){
                        for(int k=j+1;k<5;k++){
                            if(sn[k]==mem[mn+2]){
                                ans=0;
                            }
                        }
                    }
                }
            }
        }
    }

    return(ans);
}

int ch6(int *mem3 , int *mem5 ,int c3 ,int c5 , int i, int j, int k, int l,int m,int n){
    int sn[6]={i,j,k,l,m,n};
    int ans=1;

    for(int mn=0;mn<c3;mn+=3){
        for(int i=0;i<6;i++){
            if(sn[i]==mem3[mn]){
                for(int j=i+1;j<6;j++){
                    if(sn[j]==mem3[mn+1]){
                        for(int k=j+1;k<6;k++){
                            if(sn[k]==mem3[mn+2]){
                                ans=0;
```

```
                            }
                        }
                    }
                }
            }
        }

    if(ans==1){
        for(int mn=0;mn<c5;mn+=5){
            for(int i=0;i<6;i++){
                if(sn[i]==mem5[mn]){
                    for(int j=i+1;j<6;j++){
                        if(sn[j]==mem5[mn+1]){
                            for(int k=j+1;k<6;k++){
                                if(sn[k]==mem5[mn+2]){
                                    for(int l=k+1;l<6;l++){
                                        if(sn[l]==mem5[mn+3]){
                                            for(int m=l+1;m<6;m++){
                                                if(sn[m]==mem5[mn+4]){
                                                    ans=0;
                                                }
                                            }
                                        }
                                    }
                                }
                            }
                        }
                    }
                }
            }
        }
    }

    return(ans);
}

    int main()
{
    ofstream fileout;
    ifstream filein;
    int v;

    filein.open("share.txt");
    filein>>v;
    GF256 *x = new GF256[v];
    GF256 *y = new GF256[v];
    GF256 *z = new GF256[v];
    for(int c = 0; c < v; c++){
        int xi,yi,zi;
        filein>>xi>>yi>>zi;
        x[c] = GF256 (xi);
        y[c] = GF256 (yi);
        z[c] = GF256 (zi);
    }
    filein.close();

    int *mem3 = new int[105];
    int *mem5 = new int[105];

    cout<<"入力されたシェアは"<<v<<"個です"<<endl;
    for(int k = 0 ;k < v ;k++){
        cout<<k+1<<"個目のシェアの位置は"<<x[k]<<","<<y[k]<<endl;
    }

    int c3=0;
    int c5=0;

    if(v>=3){
```

```
    matrix<GF256> p(3);
    for(int i = 0; i < v; i++){
        p.e(0,0)=x[i];
        p.e(0,1)=y[i];
        p.e(0,2)=0;
        for(int j = i+1; j < v; j++){
            p.e(1,0)=x[j];
            p.e(1,1)=y[j];
            p.e(1,2)=0;
            for(int k = j+1; k < v; k++){
                p.e(2,0)=x[k];
                p.e(2,1)=y[k];
                p.e(2,2)=0;
                if(p.rank()==1){
                    mem3[c3*3]=i;
                    mem3[c3*3+1]=j;
                    mem3[c3*3+2]=k;
                    c3++;
                    t3(x[i],y[i],z[i],x[j],y[j],z[j],x[k],y[k],z[k]);
                }
            }
        }
    }
}

if(v>=5){
    matrix<GF256> p(5);
    for(int i = 0; i < v; i++){
        p.e(0,0)=x[i]*x[i];
        p.e(0,1)=x[i];
        p.e(0,2)=x[i]*y[i];
        p.e(0,3)=y[i];
        p.e(0,4)=y[i]*y[i];
        for(int j = i+1; j < v; j++){
            p.e(1,0)=x[j]*x[j];
            p.e(1,1)=x[j];
            p.e(1,2)=x[j]*y[j];
            p.e(1,3)=y[j];
            p.e(1,4)=y[j]*y[j];
            for(int k = j+1; k < v; k++){
                p.e(2,0)=x[k]*x[k];
                p.e(2,1)=x[k];
                p.e(2,2)=x[k]*y[k];
                p.e(2,3)=y[k];
                p.e(2,4)=y[k]*y[k];
                for(int l = k+1; l < v; l++){
                    p.e(3,0)=x[l]*x[l];
                    p.e(3,1)=x[l];
                    p.e(3,2)=x[l]*y[l];
                    p.e(3,3)=y[l];
                    p.e(3,4)=y[l]*y[l];
                    for(int m = l+1; m < v; m++){
                        p.e(4,0)=x[m]*x[m];
                        p.e(4,1)=x[m];
                        p.e(4,2)=x[m]*y[m];
                        p.e(4,3)=y[m];
                        p.e(4,4)=y[m]*y[m];
                        int r = p.rank();
                        if(r==4){
                            if(ch5(mem3,c3,i,j,k,l,m)==1){
                                if(t5(x[i],y[i],z[i],x[j],y[j],z[j],x[k],y[k],
                                z[k],x[l],y[l],z[l],x[m],y[m],z[m])==1){
                                    mem5[c5*5]=i;
                                    mem5[c5*5+1]=j;
                                    mem5[c5*5+2]=k;
                                    mem5[c5*5+3]=l;
                                    mem5[c5*5+4]=m;
                                    c5++;
                                }
```

```
                            }
                        }
                    }
                }
            }
        }
    }

    if(v>=6){
        matrix<GF256> p(6);
        for(int i = 0; i < v; i++){
            p.e(0,0)=x[i]*x[i];
            p.e(0,1)=x[i];
            p.e(0,2)=x[i]*y[i];
            p.e(0,3)=y[i];
            p.e(0,4)=y[i]*y[i];
            p.e(0,5)=1;
            for(int j = i+1; j < v; j++){
                p.e(1,0)=x[j]*x[j];
                p.e(1,1)=x[j];
                p.e(1,2)=x[j]*y[j];
                p.e(1,3)=y[j];
                p.e(1,4)=y[j]*y[j];
                p.e(1,5)=1;
                for(int k = j+1; k < v; k++){
                    p.e(2,0)=x[k]*x[k];
                    p.e(2,1)=x[k];
                    p.e(2,2)=x[k]*y[k];
                    p.e(2,3)=y[k];
                    p.e(2,4)=y[k]*y[k];
                    p.e(2,5)=1;
                    for(int l = k+1; l < v; l++){
                        p.e(3,0)=x[l]*x[l];
                        p.e(3,1)=x[l];
                        p.e(3,2)=x[l]*y[l];
                        p.e(3,3)=y[l];
                        p.e(3,4)=y[l]*y[l];
                        p.e(3,5)=1;
                        for(int m = l+1; m < v; m++){
                            p.e(4,0)=x[m]*x[m];
                            p.e(4,1)=x[m];
                            p.e(4,2)=x[m]*y[m];
                            p.e(4,3)=y[m];
                            p.e(4,4)=y[m]*y[m];
                            p.e(4,5)=1;
                            for(int n=m+1;n<v;n++){
                                p.e(5,0)=x[n]*x[n];
                                p.e(5,1)=x[n];
                                p.e(5,2)=x[n]*y[n];
                                p.e(5,3)=y[n];
                                p.e(5,4)=y[n]*y[n];
                                p.e(5,5)=1;
                                int r = p.rank();
                                if(r==6){
                                    if(ch6(mem3,mem5,c3,c5,i,j,k,l,m,n)==1){
                                        t6(x[i],y[i],z[i],x[j],y[j],z[j],x[k],y[k],z[k],
                                        x[l],y[l],z[l],x[m],y[m],z[m],x[n],y[n],z[n]);
                                    }
                                }
                            }
                        }
                    }
                }
            }
        }
    }

    delete []x;
```

```
    delete []y;
    delete []z;
    delete []mem3;
    delete []mem5;
}
```